IAM USER MANUALVERSION 8.0

INNOVATION DATA PROCESSING is pleased to present the IAM User Manual.

PURPOSE OF THE GUIDE

This Users Guide outlines capabilities, describes the basic features of these capabilities and provides direction on their use. It was written primarily for technical personnel who have responsibilities related to the design and implementation of systems which will use IAM. This manual is intended to give those personnel a reference that will help them use IAM as a simple, efficient, reliable alternative to VSAM KSDS and ESDS processing in batch and online applications.

IAM®, FDR®, DSF®, CPK® and ABR® are a proprietary program products of:
INNOVATION DATA PROCESSING
Innovation Plaza
275 Paterson Avenue
Little Falls, New Jersey 07424-1658

It is available for license exclusively from INNOVATION DATA PROCESSING, INC

EUROPEAN OFFICES:

France

Innovation Data Processing S.A.R.L.
191 Avenue Aristide Briand
94230 CACHAN
Tel: 01-49-69-94-02 Fax: 01-49-69-90-98
E-mail: frsupport@fdrinnovation.com
frsales@fdrinnovation.com

Netherlands

Innovation Data Processing
Brouwerstraat 8
1315 BP Almere
Tel: 036-534 1660 Fax: 036-533 7308
E-mail: nlsupport@fdrinnovation.com
nlsales@fdrinnovation.com

Germany

Innovation Data Processing
International Ltd.
Orleansstrasse 4a
D-81669 München
Tel: 089-489 0210 Fax: 089-489 1355
E-mail: desupport@fdrinnovation.com
desales@fdrinnovation.com

United Kingdom

Innovation Data Processing Ltd.
Clarendon House
125 Shenley Road
Borehamwood, Herts, WD6 1AG
Tel: 0208-905 1266 Fax: 0208-905 1428
E-mail: uksupport@fdrinnovation.com
uksales@fdrinnovation.com

A copy of this manual is provided on CD-ROM in IBM BookManager format for online use, and in Adobe PDF format for local printing

US CONTACTS:

Voice: 973-890-7300 Fax: 973-890-7147
Home page: www.innovationdp.fdr.com
E-mail: Sales: sales@fdrinnovation.com
Technical Support: support@fdrinnovation.com

THIS PAGE WAS INTENTIONALLY LEFT BLANK

TABLE OF CONTENTS

Page No.

TABLE OF CONTENTS

Page No.

TABLE OF CONTENTS

Continued

00.01 SUMMARY OF MODIFICATONS AND ENHANCEMENTS V8.0

SINGLE SYSTEM RECORD LEVEL SHARING

IAM Version 8.0 introduces the first phase of IAM Record Level Sharing (RLS) support, which features single system record level sharing capabilities. Customers can use IAM Record Level Sharing to run multiple batch jobs and CICS regions that concurrently update the same IAM enhanced format file(s) on a single z/OS or OS/390 system image. IAM RLS provides full data integrity for concurrent users of IAM files under IAM RLS control, along with journaling and file recovery capabilities. Locking is provided at the record level by IAM RLS, to prevent concurrent updating of the same record by different jobs or CICS transactions. For most applications, no JCL or programming changes will be necessary to use IAM RLS. Some of the key features of IAM RLS are:

- Record level locking across multiple jobs and CICS regions on the same z/OS image
- Provides journaling and recovery capabilities
- Automatic deadlock detection, within the realm of locks held for IAM datasets
- Supports enhanced format IAM KSDS and ESDS files
- Datasets are selected automatically for inclusion in IAM RLS, based on the file's share options and / or on installation provided list of dataset names
- Offers a batch syncpoint process for batch applications that update a substantial portion of IAM files. (Requires changes to the application program to use this facility.)
- Provides an optional Dynamic Job Backout facility, which will backout updates made to IAM files being processed under IAM RLS control, when a batch job step abends.
- Customers can select the action that IAM RLS will take when there is record lock contention. The choices are to wait until the lock is available, wait for a specified amount of time, or to immediately fail the request. If the request is failed, customers can select to either receive VSAM logical error codes and have the application programs handle the error, or request that IAM abend the job step or transaction.

Full Sysplex record level sharing capabilities for IAM files will be available in a future release.

LARGE DEVICE SUPPORT

IAM Version 8.0 provides support for IAM enhanced format datasets being defined as DFSMS extended format sequential datasets. These IAM datasets can now exceed the 64K tracks per device limitation that previously existed, allowing IAM to fully utilize 3390-9 and other even larger devices. When defined as DFSMS extended format, IAM datasets can use up to 123 extents per volume, can use all available space on the large volumes, and can exceed the prior limit of a total of 255 extents when multiple volumes are specified. This will raise the theoretical maximum file size to almost 3 terabytes of compressed data.

HARDWARE COMPRESSION SUPPORT

IAM Version 8.0 provides customers with the option of using the z/OS hardware compression. With the performance improvements that IBM has made on the data compression instructions on the zSeries processors, IAM customers may realize some CPU time reductions when reading data by using hardware compression. The use of hardware compression is not recommended on the older processors, because the additional CPU time is substantial. Hardware compression is an option that users can select for the desired files through the IAM Override facility. IAM is providing one compression dictionary, which was designed to handle primarily text data, with some numerical data, such as might be found in files containing name and address information. Users can also create their own compression dictionaries to be used by IAM for selected files. Such dictionaries can provide for a higher degree of compression than would be possible with a generic dictionary. Instructions are provided in the manual for creating a hardware compression dictionary that can be used by IAM. When an IAM file is loaded with a user provided compression dictionary, that dictionary will be written into the IAM file to insure that the dataset can be successfully processed should some subsequent changes be made to the user provided dictionary.

00.01 CONTINUED . . .

PAV SUPPORT IAM Version 8.0 provides full Parallel Access Volume (PAV) support for enhanced format IAM files. PAV is a feature of some DASD devices, which provides for concurrent physical I/O activity against the same physical z/OS DASD volume. Use of this capability can provide for substantial improvements in response times for online systems, or batch jobs, that have heavy concurrent I/O activity. Prior to IAM Version 8.0, parallel access was automatically being handled for input requests to IAM files, but concurrent output I/O requests would only be concurrently processed if the requests were to different DASD extents. With Version 8.0, changes have been made to the IAM I/O driver for enhanced format files that will facilitate concurrent output and update I/O through PAV.

BWO SUPPORT IAM Version 8.0 provides support for users of the callable BWO (Backup While Open) service. This interface is primarily used by CICS and CICS/VR to allow for backing up open VSAM files, and then for subsequent forward recovery by CICS/VR after the dataset has been restored. IAM will now keep the BWO data in the IAM file's control information. The BWO data includes three flags, indicating the BWO state of the file, and the 8-byte date and time stamp field. The date and time stamp field is used as a file recovery starting point in time by CICS/VR when recovering a dataset that has been restored. When present in an IAM file, the BWO data will be printed on the IAMPRINT LISTCAT report.

IAM files could previously be backed up while open, and the restored copy would retain file structure integrity. However, now with the IAM BWO support, the forward recovery function can automatically determine the point in time from which the recovery should start. In effect, IAM BWO callable service support allows for the recovery coordination between CICS and CICS/VR.

IAM JOURNAL RECOVERY IAMJREST is the IAM program that will read data from an IAM journal, and perform the requested recovery. For V8.0, this program has been enhanced to handle the journal datasets produced by IAMRLS, and perform either a forward or backward recovery, as requested. This enhancement includes new keywords and functionality for recovering files after abends or failures when such files are used under IAM RLS. If a batch job abends, the IAM datasets that have been accessed through IAM RLS, can have the changes removed from the IAM file by a process referred to as a backout. The backout can be for only the failing job step, for the entire job, or also for other preceding jobs. If the failing program makes use of the IAM Batch Syncpoint, the back out can be done up to the last syncpoint taken prior to the abend.

Also note that all of the utility functions previously performed by IAMJREST have been moved to a new program, called IAMJUTIL. IAMJREST will only handle the actual recovery. To print journal records, find out information from journal records, or resetting the System Logger files, IAMJUTIL can be used.

AIX USABILITY ENHANCEMENT Many users of IAM/AIX have been taking instant backups or snapshot copies of IAM base clusters without the alternate indices or paths. This has caused some various problems because the IAM AIX information is still in the copied base cluster. With V8.0, the IAM/AIX alternate index information can be removed from the copied base cluster through an IDCAMS delete command that specifies the copied base cluster name, and the keywords AIX and NOSCRATCH. A subsequent LISTCAT on the file will verify the removal of the AIX relationships from the base cluster.

Z-SERIES PROCESSORS IAM Version 8.0 has modified various portions of code that may have been adversely impacted by changes in the cache structure on the IBM Z-series processors. This may result in some CPU time reductions for customers upgrading from prior releases of IAM. The modifications were made in such a manner as to not degrade IAM performance for customers on the prior IBM mainframe processors.

VSAM QUICK INDEX The VSAM Quick Index from EMC (formerly Softworks) can be used with IAM alternate indexes starting with V8.0 of IAM. VSAM Quick Index must be run either defaulting to, or explicitly specified on the BASE control card, the NOFASTPATH option. Customers that are going to have both V7.0 and V8.0 of IAM active on the same LPAR during the V8.0 test period must apply PTF P-70.0150 to their V7.0 IAM to use VSAM Quick Index with IAM V8.0.

00.01 CONTINUED . . .

IAM ALTERNATE INDEX PERFORMANCE

AIX PERFORMANCE As can be seen from the example data provided below, IAM can provide significant performance benefits over using VSAM. The performance benefits of using IAM for any particular application will vary from the results shown here, as there are many variables. Such variables include the number of upgradeable alternate index datasets, the data reference pattern, the amount of VSAM tuning that has been done, plus many others. The jobs were run on a Z800-2066 CPU under z/OS 1.3.

The following numbers are from a base cluster with 2,500,000 records with two upgradeable alternate index datasets. One of the alternate index datasets has UNIQUE keys, the other has NONUNIQUE keys, and both were upgradeable. All of the measurements were done with minimal other activity on the system.

The IAM run was performed with the default options and the following buffer overrides:

```
ACCESS  DD=&ALLDD,MINBUFNO=60,MAXBUFNO=100
```

Three different VSAM runs were done, one with no buffering specified, one with DFSMS System Managed Buffering (SMB), and one with the VSAM files in using an explicit LSR buffer pool. The LSR pool had 256 data buffers, and 384 index buffers.

00.01 CONTINUED . . .

RANDOM UPDATE I/O TEST

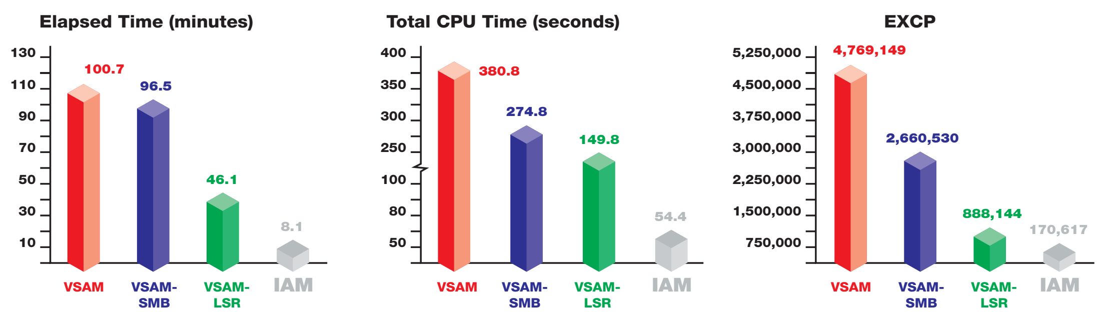
This job step performed a variety of read and update requests through one of the alternate index datasets. The following type of I/O requests were issued against the path:

- 256,000, Random Read
- 160,000 Updates, 1/2 of which required an update to the other alternate index.
- 80,000 Inserts
- 16,000 Deletes.
- 16,001 Points (Start browses) with 80,000 sequential reads. (5 records read per start browse).

The results were:

- **IAM used 64% to 84% less CPU time than VSAM.**
- **IAM performed 81% to 96% less EXCP's than VSAM.**
- **IAM ran in 82% to 92% less Elapsed Time than VSAM.**

The performance data is from the SMF data for the jobs that were run. The total CPU time figure includes both TCB and SRB time



IAM took 8 min. versus 46 min. for VSAM tuned with LSR

	IAM	VSAM	VSAM – SMB	VSAM – LSR
Total CPU Time (seconds)	54.4	380.8	274.8	149.8
EXCP	170,617	4,769,149	2,660,530	888,144
Elapsed Time (minutes)	8.1	100.7	96.5	46.1

Figure 1: Results of Random I/O Performance Test

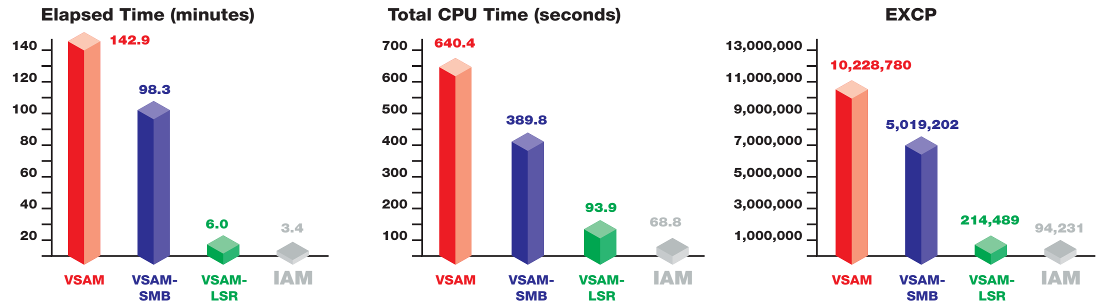
00.01 CONTINUED . . .

SEQUENTIAL I/O TEST

After updating the files with the above Random I/O test, a second test was performed. This test read the file sequentially using the NONUNIQUE key alternate index.

The results were:

- **IAM used 27% to 89% less CPU time than VSAM.**
- **IAM performed 56% to 99% less EXCP's than VSAM.**
- **IAM ran in 43% to 98% less Elapsed Time than VSAM.**



IAM took 3 min. versus 6 min. for VSAM tuned with LSR

	IAM	VSAM	VSAM – SMB	VSAM – LSR
Total CPU Time (seconds)	68.8	640.4	389.8	93.9
EXCP	94,231	10,228,780	5,019,202	214,489
Elapsed Time (minutes)	3.4	142.9	98.3	6.0

Figure 2: Results of Sequential I/O Test

00.02 SUMMARY OF MODIFICATIONS AND ENHANCEMENTS IAM V7.0

AIX SUPPORT IAM Version 7.0 has an optional feature available for an additional cost, of Alternate Index Support. Customers with the IAM AIX option can define and use alternate index data sets and paths to enhanced format IAM KSDS and IAM ESDS types of files. VSAM data sets that previously had to be excluded from use of IAM due to having one or more alternate index data sets can be converted to IAM. This optional feature will bring the significant performance benefits and IAM data compression to those data sets. IAM also provides support for over 4 gigabytes of data within a data set, which may be a major benefit to customers that are not using DFSMS to manage their VSAM data sets. No changes are required to application programs or CICS to use the IAM alternate index data sets. As a general rule, no JCL changes are required either. Just simply indicate on the IDCAMS define of the base cluster that the data set is to be an IAM data set. Some changes to data set management utility processing may be necessary to properly handle IAM alternate index data sets, particularly if an IAM data set that is part of an alternate index association is renamed.

Whenever a define of an alternate index or path is done, IAM will examine the request to see if the related data set is an IAM data set. If it is, then IAM will handle the file definition, otherwise normal VSAM processing continues. The IAM alternate index is created as an enhanced format IAM KSDS type of file. The record format within the IAM alternate index is fully compatible with the VSAM format. IAM alternate index data sets can be built by either the IDCAMS BLDINDEX function, or by any program designed to create a VSAM alternate index. IAM provides complete support for alternate indexes defined with either UNIQUE keys or NONUNIQUE keys. IAM also provides complete support for upgradeable alternate index data sets.

The IAM path is created as a one-track data set to contain the necessary association information. IAM provides support for paths on alternate index data sets and on the base cluster itself. IAM honors the upgradeable attribute of the path.

The alternate index support has required changes to the following functions of IAM:

- DEFINE alternate index and paths, and DEFINE RECATALOG requests.
- File load support for alternate index data sets.
- New Open, Close, and logical I/O request processing modules to handle I/O activity using alternate index to access a base cluster, and to provide the alternate index upgrade support.
- Enhanced SHOWCAT and catalog LOCATE support for alternate index data sets and paths, plus support of the associations.
- Enhancements to List Cat (IAMPRINT) report for alternate index data sets and paths.
- Delete support for IAM data sets with associated alternate index and paths.
- IAM ISPF panels to support utility functions.

SPANNED RECORD SUPPORT IAM Version 8.0 supports spanned records. This capability allows record sizes that exceed the amount of data that can be kept in a single block. A spanned record in an IAM file can now reside in up to 256 data blocks. This support will eliminate the DASD space that is wasted for files that have a maximum record length that exceeds the half-track block size that previously forced IAM to use a block size of 32760. IAM can now support larger record sizes than IAM was previously able to handle. Spanned record support will also allow customers to have alternate index files that may need to have many records with the same alternate key. To enable spanned record support for a file, simply specify the SPANNED attribute when the file is defined. Please be aware that while IAM will support record sizes in excess of 32760 bytes, many utility programs, including IDCAMS and the SORT packages do not support records in excess of 32760 bytes.

00.02 CONTINUED . . .

ADDITIONAL ENHANCEMENTS TO IAM V7.0

IAMOVRID ENHANCEMENTS

A new keyword, **DSN=**, has been added to the CREATE override for both compatible and enhanced format files and to the ACCESS override for enhanced format files. With this enhancement, data sets can be identified by either data set name or DD name on the IAM override card. Additionally, the maximum value that can be specified for **MINBUFNO** and **MAXBUFNO** has been increased to 2048.

TEST VIF ENHANCEMENT

The VIF testing capability has been enhanced to support up to 5 different job names or job name prefixes. In prior versions, the user could only have one job name or job name prefix active at any time.

IAM ISPF PANEL ENHANCEMENT

In addition to the new support in IAM ISPF for alternate index data sets and paths, the copy function now supports sequential files. This means that either an IAM or VSAM data set can be copied into a sequential data set, or that a sequential data set can be copied into an IAM or VSAM data set. Also, the IAMXMON facility for monitoring I/O activity of IAM enhanced format files can now be invoked through the IAM ISPF panels for CICS regions running under the same CPU or LPAR.

NEW IAM GLOBAL OPTION

Two new IAM Global Options, **CICSBUFSP=** and **REORGWTO**, have been added for IAM V7.0. The **CICSBUFSP** parameter specifies the default maximum amount of storage to use for buffer space when IAM files are used under CICS. This will help situations where customers would prefer different defaults for batch buffering versus online buffering. The default value is 262,144 (256K). The **REORGWTO** provides the capability for customers to turn off the IAMW22 message that is an informational message suggesting that particular IAM files should be reorganized.

CHANGED GLOBAL OPTIONS

The following Global Option default values have been changed for IAM Version 7.0. Please review them carefully, particularly if you are upgrading from IAM Version 6.3 or earlier.

- **The BUFSP value has been increased to 875K (896,000).** This will allow IAM to set MAXBUFNO to a value large enough to hold slightly more than one cylinder of data. The value had previously defaulted to 256K. CICS will use the new Global Option CICSBUFSP for this calculation.
- **The CRBUFOPT default has been changed to MCYL.** This will automatically provide the best buffering possible for file loads. The value previously had defaulted to CYL.
- **The DATASPACE value has been increased to 256 megabytes.** This change was done to allow for larger Index Space storage to be obtained. The previous default was 128 megabytes.
- **The LOADABWO option is now defaulting to YES.** A value of YES is required for customers running CICS Transaction Server. Customers running earlier levels of CICS should change this back to NO.
- **The MAXREGION default value has been increased to 512 megabytes.** The value had previously been set to 128 megabytes. The increase was made because many customers already had CICS regions in excess of 128 megabytes, which eliminated the benefit of this feature.
- **The VAROVERFLOW default has now been set to YES.** Customers that are upgrading to this version directly from Version 6.3 should change this to the prior default of NO, until this version of IAM has been accepted in a production environment. This is because use of variable overflow is not downward compatible with Version 6.3 Enhanced format files.

00.03 CONTROL STATEMENT FORMAT

GENERAL The IAM control statements consist of 80-character logical records. The general format of these records is:

```
Column                                                      Column     Column
  1                                                           72         80
-------------------------------------------------------------------------
  |                                                         | |        | |
  |    COMMAND           OPERANDS          COMMENTS         | |        | |
  |                                                         | |        | |

WHERE:

Columns 1 to 71 – contain the command, operands, and comments fields,
except when continued to  subsequent logical records.

Column 72 – must be blank if the command or operands extend to column 71.

Columns 73 to 80 – not used by IAM. We suggest you use them as an identification
or sequence field.
```

Figure 3: General Control Statement Format

COMMAND FIELD The command field identifies the control statement and consists of a one or more character command word. It may appear anywhere within columns 1 to 71 and may only be preceded by blanks. The command word must appear in its entirety within columns 1 to 71; it may not be continued.

OPERAND FIELD The operand field, if present, follows the command field and is separated from it by at least one blank. The operand field consists of one or more keywords and/or positional parameters, separated by commas. It may not contain embedded blanks except within quoted strings. Operand fields may be continued onto subsequent logical records. If operands are to be used with a command, at least one operand must be on the logical record containing the command.

COMMENTS FIELD The comments field, if present, follows the operand field(s) and is separated by one or more blanks. It may contain any information deemed helpful by the person who codes the control statement. Comments fields may not be continued, i.e., they must end at or before column 71. Comments are not permitted on a control statement that allows operands but on which no operands have been specified.

CODING AN OPERAND FIELD An operand field consists of one or more positional or keyword parameters separated by commas.

POSITIONAL PARAMETERS Positional parameters must be coded in a specific order relative to one another. This means that the variable data you substitute for positional parameter 1 must precede the variable data for positional parameter 2, and so on. The absence of variable data to be substituted for a positional parameter is indicated by coding a comma in its place. However, you may omit the commas when the absent parameter is the last one, or if all following positional parameters are absent.

KEYWORD PARAMETERS Keyword parameters are position independent, and consist of either a keyword alone or a keyword followed by an equal sign (=) followed by user-specified variable information. When both positional and keyword parameters are to be coded in an operand, keyword parameters must precede positional parameters.

SUBPARAMETERS Both positional and keyword parameters may consist of a list of subparameters. Such a list is composed of positional parameters that follow the usual rules for that type. A subparameter list must be enclosed within parentheses, unless the list reduces to a single subparameter, in which case the parentheses may be omitted.

00.03 CONTINUED . . .

QUOTED STRINGS

When the variable data you specify for a parameter contains certain special characters, defined below, you must enclose the data with apostrophes. This is called a 'quoted string'. Within a quoted string, all characters, including blanks, may appear; if an apostrophe is to be part of the string, it must be coded as two apostrophes.

NOTE: Special characters are parentheses, commas, equal signs, apostrophes, and blanks.

CONTINUING AN OPERAND FIELD

When the total length of an operand field exceeds the available columns in a logical record, it must be continued onto one or more following logical records. To continue an operand, interrupt the operand field after a complete parameter or subparameter, including the following comma. Do this at or before column 71. Leave the next column blank.

Continue the operand field starting anywhere between columns 1-71 on the following logical record.

Any number of logical record continuations may be present.

COMMENT STATEMENT

If you really have a lot to say, you may code a comment statement, which are control statements with nothing but remarks contained within columns 2 to 71. Comment statements are identified by an asterisk (*) in column 1. They may appear anywhere within a group of control statements, even between continued logical records.

EXAMPLE CONTINUED STATEMENT

```
REPORT   MAXDSNS=6000,                EXAMPLE OF A
         DETAIL,                      CONTINUED OPERAND
         GROUPNAMES=(CICS,ABC),       FIELD, WITH
         MAXREPORTS=5000              COMMENTS
```

Figure 4: Example of a Continued Control Statement

00.04 DOCUMENTATION NOTATION FORMAT

NOTATION The following notation is used in this manual to define control statement formats:

- **Uppercase letters and words** must be coded exactly as shown in a format description.
- **Lowercase letters and words** represent variables for which you must substitute specific information.
- **Brackets** are never coded. They indicate that the enclosed item is optional, and you can code one or none of the items.
  For Example: **[,MAXDSNS=nnnn]**
- **An ellipsis . . .** (3 consecutive periods) is never coded. It indicates that the preceding item can be coded more than once.
  For Example: **DSN=(dsname,dsname . . .)**
- **An underscore** ______ is never coded. It indicates that the underscored item is the abbreviation for the value.
  For Example: **<u>DATA</u>COMPRESS=** YES | NO
- **A vertical bar** | is never coded. It indicates that there is a choice between the values specified.

01.01 IAM VERSION 8.0 INTRODUCTION

WHAT IAM IS IAM is a reliable high performance disk file manager that can be used in place of VSAM KSDS or VSAM ESDS datasets for batch, TSO and online processing. Customers with the optional Alternate Index feature can use IAM for alternate indexes and the associated base clusters. IAM offers a level of performance and reduction in the use of computer system resources that provides substantial savings for most applications that utilize VSAM datasets. IAM drastically reduces DASD space requirements through maximizing the storage capabilities of each DASD device and a Data Compression feature that does not consume excessive CPU time. IAM utilizes a sophisticated buffering mechanism, called Real Time Tuning that responds quickly to the I/O demands of application programs, resulting in substantial decreases in physical I/O.

IAM FEATURES Because of IAM's innovative file structure and software design, IAM offers the following capabilities:

- A transparent VSAM interface that requires no changes to application programs, and generally no changes to JCL or CICS regions.
- Support for base KSDS and ESDS files, and as an option, any associated Alternate Indexes. With IAM's PSEUDORBA feature, many applications can use IAM ESDS files that exceed 4 gigabytes without having to change their applications to handle the 8-byte RBA that VSAM requires.
- Potential for up to three terabytes of compressed data within a single IAM dataset when IAM datasets are defined in DFSMS extended format.
- Data Compression facility that can be used for both KSDS and ESDS types of datasets, without the significant CPU overhead that is typical of most data compression techniques. IAM Version 8.0 offers both an efficient proprietary software data compression technique or on zSeries processors customers can opt to use hardware compression. Using hardware compression with customized compression dictionaries, users may be able to achieve even greater data compression than can be achieved with standard VSAM hardware compression. Data compression also helps to reduce physical I/O required to process the datasets, because more data is contained in a physical block than when it is uncompressed.
- IAM's automatic Real Time Tuning offers dynamic buffer management techniques that are unsurpassed. <u>The need for manual tuning is eliminated for almost all files.</u>
- Record level sharing within a single OS/390 or z/OS image, for concurrently executing batch jobs, CICS regions and TSO users.
- Automatic space release after a file is initially loaded and dynamic secondary space adjustments.
- Data compressed files can be reorganized with FDRREORG or IDCAMS without decompressing and recompressing the data. This reduces CPU time and I/O to perform the reorganizations, and it can reduce media usage when the sequential copy of the data is placed on DASD.
- A "Data in Virtual" feature called Dynamic Tabling, which keeps those records that are frequently required in a virtual storage table.
- IAM's run time report IAMINFO, which fully describes each IAM dataset processed by a job step, along with a complete processing profile, including resource usage. This data can also be captured by SMF, with reporting available from programs provided with IAM, or customers can write their own SMF analysis and reporting programs to include the IAM provided data.

IAM provides this unique set of features that are not found in other individual products. There are several software packages offered by other vendors that will provide some of the above functions, but only IAM puts all of this capability together in one package.

01.10 HIGHLIGHTS OF IAM VERSION 8.0 ENHANCEMENTS

RECORD LEVEL SHARING

The IAM single system Record Level Sharing permits multiple jobs, CICS regions and /or TSO users to concurrently update IAM files, with full data integrity. Locking is performed at the record level, so the different jobs can update records within the same data block without any delay. With this capability now built into IAM, customers can come even closer to achieving continuous availability of online transaction processing systems. IAM's Record Level Sharing additionally provides for journaling file update activity, and the ability to either backout updates from failing batch jobs or TSO users, or to perform a forward recovery if an IAM file had to be restored from an earlier backup.

IAMRLS is easy to use. The setup checklist for preparing for IAM RLS includes the following:

1. Allocating an IAM RLS parameter library, and choosing the startup parameters.
2. Allocating the IAM RLS journal datasets, if journaling is going to be used, and setting up backup and empty procedures for the journal datasets.
3. Deciding on eligibility criteria for IAM RLS usage and building the dataset name table if it is going to be used.
4. Setting up the IAM RLS proc from the example provided in the IAM Installation Control Library (ICL), and placing it in a system procedure library.
5. Start the IAM RLS procedure. This can even replace the IAMSTART procedure in existence today, because starting up IAM RLS will automatically start up the IAM VSAM Interface if it is not already active.

Once IAM RLS is active, datasets will be automatically selected for processing by IAM RLS, based on installation criteria. Most application programs and jobs will not require any changes to code or JCL to make use of IAMRLS. Datasets will have to be defined as IAM datasets. The circumstances where changes may be required are:

1. For batch applications that update a large number of records on recoverable files, it is recommended that they be modified to periodically call the IAM batch syncpoint. This will aid in preventing a large number of records from being locked out for update by other jobs and CICS transactions.
2. If an installation chooses to have IAM issue a transaction abend under CICS when a record lock is not available, or after a time out while waiting for a record lock, they will have to install the IAM file control exit XFCREQC.

The IAM RLS address space is designed and intended to provide it's services in the background, with little if any operator intervention. However, should the need arise, there are a variety of commands that can be given to the IAM RLS address space through the MVS MODIFY (F) command. These include display as well as action commands. IAM RLS activity can also be viewed from a TSO/ISPF session, using the provided IAMBMON program. This program can be invoked either as a TSO command from any ISPF panel, or from the IAM ISPF utility menu.

01.10 CONTINUED . . .

IAM RLS PROCESSING OVERVIEW

The IAM Record Level Sharing within a single system is achieved by IAM internally passing all of the I/O requests for the files being shared to the IAM RLS address space. This is done automatically for files that meet the installation's selection criteria, which can be determined from a combination of share options and dataset names. If the IAMRLS address space has not been activated, then processing for those files will be performed as it is currently within the job's address space. The use of IAMRLS can also be controlled by use of the IAM ACCESS overrides of RLS and NORLS.

IAM utilizes z/OS cross memory services to provide the record level sharing services. All of the OPEN, CLOSE, and I/O requests for shared files will be passed to the IAMRLS address space for processing, and the appropriate status, key and record will be passed back to the user's address space when the request has completed within the IAM RLS address space. Record level locking is achieved within the IAMRLS address space, with a lock manager that will handle the locking requests, and provides for deadlock analysis and detection within the resources managed by IAM.

IAM V8.0 also utilizes the z/OS dynamic resource manager facilities so that IAM will get control whenever a job step or an address space that is using IAM files through IAMRLS terminates. This function will make sure that the IAM file(s) are properly closed within the IAMRLS address space, will release any record locks for jobs that are normally ending, and will retain record locks of recoverable files for jobs that are abending.

JOURNALING

In addition to handling the I/O and locking services, the IAMRLS address space can also journal before and / or after images of updated records. The journaling can be performed either to standard DASD sequential datasets, or to the z/OS System Logger. The IAM journaling services are primarily provided to allow for the back out of updates performed by failing batch job(s), or to perform a forward recovery of updates if a file has encountered media damage. CICS will handle it's own transaction back out and other recovery as it does today, using it's own logging mechanisms that is independent of the journaling provided by IAM.

SECURITY

The IAMRLS address space will have to be given security authority to update those files that are going to be processed by IAM record level sharing. IAM does issue the RACROUTE macro within the individual job's address space to validate that the requesting user does have authority to read or update, as appropriate, the IAM dataset being opened prior to requesting that IAMRLS open the dataset. If the RACROUTE indicates that the users/job does not have authority to access the dataset, then the OPEN request is failed. The failing job will receive an IAMW18 error message.

RELIABILITY

IAMRLS utilizes the various z/OS error handling and recovery facilities to recover from errors and abend conditions that may occur. Our two goals in providing error recovery routines are to provide continuous availability of the services being provided by the IAMRLS address space, and secondly to automatically collect enough information about any failures such that problem determination and correction can be performed from the single failure. The job that had submitted an I/O request to IAMRLS that causes a failure will in most circumstances, receive a failure code. It will be up to the program to decide whether to continue processing.

The error data collection will include various messages to the job log, system log, and when applicable the RLS log indicating the abend code, general registers, access registers if applicable, and the failing module. The error routines do attempt to not duplicate diagnostic information that is produced by z/OS, but rather provide additional diagnostic information to be combined with the information provided by z/OS. The error information contained in these messages may be sufficient for problem determination and resolution, particularly if a problem had been previously reported to Innovation. If the error occurred within the IAMRLS address space, an error trace table will be kept in storage for reference, particularly in situations where multiple errors have occurred. Most error situations will also result in a request for a dump to be taken to a system dump dataset, which may include both the IAMRLS address space and possibly the address space of the job that submitted the failing request. The IAM index space associated with the IAMRLS address space may also be dumped.

SERVICEABILITY

To aid in serviceability a mechanism is being provided for system support personnel to apply critical fixes to the IAM modules within the IAMRLS address space, without the need to shut down the address space. This facility will also be able to back out fixes should they cause other problems.

01.10 CONTINUED . . .

RECORD LOCKING For the single system record level sharing, IAM utilizes its own record level lock manager. The IAM lock manager utilizes a hashing algorithm to provide for fast lock acquisition and release. The IAM lock manager does check for potential deadlocks within the scope of the IAM datasets that it is managing. If by waiting on the record lock for the current request will result in a deadlock, the request will be failed with a logical error. The trigger for IAM to release a record lock will depend on the environment in which the lock was requested.

For CICS transactions, records are locked by transaction identification. IAM assumes that the IAM files accessed under CICS are recoverable, and therefore holds the record lock(s) until either a SYNCPOINT is executed, or the transaction ends. If the IAM file is defined with the explicit option of JRNAD=NONE or the IDCAMS parameter LOG(NONE), then IAM will assume that the file is not recoverable under CICS. When an IAM file is detected as being not recoverable under CICS, then the record lock is only held from the time of the GET for UPDATE until the record is actually updated or erased, or for records being added, only for the duration of the actual add processing.

For other than CICS processing, that is batch jobs or TSO users, IAM will generally only hold the record lock(s) from the time of the GET for UPDATE until the record is actually updated or erased, or for records being added, only for the duration of the actual add processing. The exception to this is if IAM is journaling before images of records, implying that a back out could be performed if there is a failure, then the record lock(s) will be held until the program either calls the IAM batch syncpoint, or the job step terminates.

If a job step abends while processing a recoverable file, then any record locks obtained for that job step will be held until a recovery takes place. If there were any other jobs or CICS transactions waiting for the record locks that are being retained until recovery, those requests will be failed when the abend has been detected. The recovery can be performed by IAM's Dynamic Job Back Out function, by IAMJREST, or by whatever other procedures or recovery software that may be available. Information about any retained locks can be found by using the DISPLAY,RETAINEDLOCKS modify command to IAMRLS. Such retained locks can also be released by the RELEASELOCKS command.

DYNAMIC JOB BACK OUT IAM V8.0 also provides a Dynamic Job Back Out function for IAM files opened through IAM RLS. Whenever a job step abends, all updates done by that job step will be automatically removed. If the failing job step has taken IAM batch syncpoints, then the back out will be performed to the most recent syncpoint taken by the job step prior to abending. Control of Dynamic Job Back Out will be provided as an IAM RLS parameter, and as an IAM Override.

LARGE DEVICE SUPPORT In prior releases, IAM files were limited as a standard non-VSAM dataset to a maximum of 65,535 tracks per volume. In IAM V8.0, enhanced format IAM files may now be optionally defined as DFSMS extended format sequential datasets. These datasets are now able to fully utilize larger volumes, such as a 3390-9 or even larger volume of up to 32K cylinders. These datasets can use up to 123 extents per volume and exceed the prior limit of 255 extents in total when multiple volumes are specified. Such datasets must be DFSMS managed, with a data class that specifies "Extended Required" or "Extended Preferred" for the dataset name type, and with Compaction set to NO. These datasets must also have a Storage Class with a "Sustained Data Rate (SDR) either left blank, or set to 0. Setting other values for Sustained Data Rate will cause the dataset to have multiple STRIPES, which IAM does not support. IAM will internally compress the file, with either software or hardware compression.

There are a couple of disadvantages of using DFSMS extended format datasets. First, DFSMS automatically adds 32 bytes of control information to each block of data written out to DASD. Because of this additional overhead, the extended format datasets may require slightly more DASD space than an IAM dataset that is not extended format. Secondly, IAM will use BSAM to write out the blocks to the file when extended format datasets are loaded. While IAM will cause BSAM to write out multiple blocks per physical I/O, the EXCP count for the file load with BSAM will be the number of blocks written out to the dataset, not the actual number of physical I/Os. With BSAM, the actual number of physical I/Os is not known, and is subject to variance from one run to another. (**Note:** This may cause the IAM EXCP count to be higher than when loading a VSAM dataset, because VSAM is counting the actual number of physical I/O's, not the number of control intervals written out.) Therefore, it is recommended that extended format datasets only be used when the capabilities provided by this format is needed.

01.10 CONTINUED . . .

PARALLEL ACCESS VOLUME (PAV)

Parallel Access Volume (PAV) support is automatically provided for enhanced format IAM files. This capability will allow concurrent physical I/Os on the same volume. Use of this capability can provide for substantial improvements in online systems, or batch jobs, that have heavy concurrent I/O activity. While Parallel Access Volume support was previously automatically used for some of the I/O in prior versions of IAM, with Version 8.0 changes were made to the IAM I/O driver to facilitate PAV usage for all possible I/O, in particular for concurrent output and update I/O.

HARDWARE COMPRESSION

With the zSeries processors, IBM has significantly reduced the CPU time for the hardware compression and decompression instruction. Because of this, IAM is now providing an option to use the IBM hardware compression instead of the IAM software compression for selected files. One of the advantages of the IBM hardware compression is that it has the potential to achieve significantly better compression for some datasets than other compression techniques. The basis of the IBM hardware data compression is that it utilizes a compression dictionary that contains frequently observed strings of data within the source dataset. These common strings of data are then compressed down into an index into the decompression dictionary. So the more representative the dictionary is of the actual data in the dataset, there will be a greater amount of compression.

IAM in Version 8.0 is providing one default hardware compression dictionary, which is oriented towards commonly used strings of data in the English language, and also within a name and address type of data. The dictionary includes compression for some numeric strings. As this dictionary will not be beneficial for many datasets, the IAM manual includes instructions for customers to create their own customized dictionaries that may achieve even greater compression. The specification of a customized compression dictionary is provided when a file is defined or loaded via IAM overrides. When a customer provides their own compression dictionary, IAM will save a copy of that dictionary within the file to assure that the data can subsequently be decompressed.

To use hardware compression specify on an IAM CREATE override the keyword DATACOMP=HW. When using a customized dictionary for the hardware compression, you must also specify the IAM CREATE override DICTIONARY=nnnn, where nnnn is the four character suffix of the dictionary name. (**Note:** The first four characters of the dictionary name are required to be: IAMD)

BACKUP WHILE OPEN (BWO)

IAM Version 8.0 provides support for the Backup While Open (BWO) service interface. IAM will keep the BWO information, consisting of status flags and a time stamp. The BWO callable service will function for IAM files as it does for VSAM files, to retrieve and update the BWO data. Additionally, when BWO data is present in the IAM file control information, that information will be displayed in the IAMPRINT LISTCAT report. This will primarily be of benefit for recovery programs, such as CICS/VR, which will utilize the BWO information.

IAM OVERRIDE CHANGES

A number of the unsupported IAM override keywords had been removed from the IAM override parser in Version 7.0, resulting in errors and problems for customers that had jobs with the older keywords. Those keywords have been added back into the IAM override parser for Version 8.0, so they will not cause the override statements containing those keywords to be flagged in error.

For the IAM **CREATE** override card, there are some changes in support of the hardware compression. There is a new keyword, **DICTIONARY=nnnn** that specifies the four character suffix for the hardware compression dictionary name. There is also a new operand for the **DATACOMP** keyword, which is **HW**, which is used to indicate that IAM is to use the IBM hardware compression instruction instead of the IAM software compression.

For the IAM **ACCESS** override card, there are some new keywords in support of the Record Level Sharing function. There are the **RLS** and **NORLS** keywords that will override the automatic decision on whether or not to use RLS for the indicated files. If RLS is being used, the other new keyword is **DJB=[YES|NO]** to indicate whether or not IAM can use Dynamic Job Backout for this file.

01.20 SAMPLE VENDOR PRODUCTS THAT HAVE CONVERTED VSAM FILES TO USE IAM

This is a sample of some of the Vendors' products that Customers have converted VSAM files to use IAM

American Software – Materials Management

American Management System – CUFS, ACAPS

Arthur Andersen – DCS

Credit Card Software – Banking Applications

Computer Applications Services – Abend Catcher

Computer Associates – CA-7, C-11, Activator, UFO, Intertest, Netman, Easytrieve+, Bundl, Netmaster

Cyborg – Payroll System

Datalink Systems – Fast Teller System

Dun & Bradstreet Software – (GEAC) Accounts Payable, General Ledger, Millennium Series Products

Dyatron – Employee Benefits

Erisco – Claim Facts

First Bankcard System – TBS, TCS

Global – CARMS

Group 1 – Code 1/PLUS, Mailsort

H&W Computer – Wizard, Mail, SYSM, SYSB2

Healthquest (HBO) – Patient Information File

Hogan Software – Demand Deposit

IBM – SMP/E, RMDS

ISS America – SICS

Impel Pacific – Power Plant Maintenance

Information Builders – Focus

Information System of America – Prism

Integral Systems – Human Resources/Payroll, HRMS

Kirchman (Florida Software) – Installment Loan, CIF, TESAREC, Dealer Floor Plan

LPC – Mailers, Finalist, Choice

MacKinney Systems – Job and Syslog Facility

Mobius – INFOPAC

Newtrend – Infopoint

Policy Management Systems – APS

SAP – RF (Financial Accounting), RV (Sales and Distribution), RM-PPS, RM-MAT

Shared Medical Systems (SMS) – Invision (Active Patient Database file)

Shaw – Installment Loan, Commercial Loan

Stockholders Systems – PEP+, CSS II

Systematics –

VIPS – Medical Part B

Vantage – Vantage 1 Annuities

Walker – General Ledger, CARMS, Materials Management

Xycor – XY-CARD, XY-CLAIMS AND XYADMIN

01.21 SAMPLE VENDOR PRODUCTS THAT WORK WITH IAM

IAM works with many system management software products that are used with VSAM today. Below is a list of products known to work with IAM. In many circumstances, these vendors have made enhancements to their products so that they can be used with IAM files. This is by no means a complete list of all the products that do work with IAM files.

JOURNALING AND RECOVERY:

FILE SAVE	(CA)
DRS	(BMC)
CICS/VR	(IBM)

SHARING PACKAGES:

SHARE OPTION 5	(CA)
SYSB	(H & W)
VSAM ACCESS/CICS	(BMC)

MISCELLANEOUS PRODUCTS:

FILE-AID	(COMPUWARE)
ABEND-AID	(COMPUWARE)
CICS	(IBM)
ISPF	(IBM)
NETWORK DATA MOVER (NDM)	(CA)
SHRINK	(CA)

DASD MANAGEMENT SOFTWARE:

FDR/ABR	(INNOVATION DATA PROCESSING)
FDRREPORT	(INNOVATION DATA PROCESSING)
FDRREORG	(INNOVATION DATA PROCESSING)
DF/SMS	(IBM)
DF/HSM	(IBM)
DF/DSS	(IBM)
VAM	(CA)
DMS/OS	(CA)
POOLDASD	(BOOLE & BABBAGE)
STOPX37	(BOOLE & BABBAGE)

SORT PRODUCTS:

SYNCSORT	(SYNCSORT)
DF/SORT	(IBM)
CA/SORT	(CA)

SECURITY PRODUCTS:

RACF	(IBM)
ACF/2	(CA)
TOPSECRET	(CA)

PERFORMANCE MONITORS:

OMEGAMON	(CANDLE)
THE MONITOR TMON	(ASG)
STROBE	(COMPUWARE)

PROGRAMMING LANGUAGES:

VS/COBOL	(IBM)
COBOL II	(IBM)
FORTRAN	(IBM)
PL/1	(IBM)
BAL	(IBM)
CA/OPTIMIZER	(CA)
SAS	(SAS INSTITUTE)

THIS PAGE WAS INTENTIONALLY LEFT BLANK

02.01 IAM CONCEPTS AND FACILITIES

IAM BASICS

IAM is a high performance indexed access method for z/OS and OS/390 operating systems, which offers numerous advantages over the IBM provided VSAM access method. IAM files exist on DASD as non-VSAM datasets, with IAM providing a VSAM compatible API (Application Programming Interface) for KSDS and ESDS file types, and, available as an option, any associated alternate indexes. Existing as non-VSAM datasets provides IAM with capabilities to eliminate the 4.3-gigabyte file size restriction in VSAM prior to DFSMS 1.3, and to choose a block size that will optimize space utilization on each of the different type of DASD devices and architectures available. Along with IAM's unique file structure, described in a subsequent section, and IAM's Data Compression feature, user data stored in an IAM file typically requires substantially less DASD space then when stored in a VSAM cluster.

The basic principle of IAM is to intelligently utilize virtual storage to reduce the need to perform physical I/O to look up and retrieve data. This is accomplished by using the sophisticated IAM Real Time Tuning concept to manage buffers, and by keeping the index for files in virtual storage while they are being processed. IAM requires at most one I/O to retrieve any explicitly identified record within the IAM dataset. All of the index I/O's and index buffers used by VSAM are eliminated.

DYNAMIC FILE EXPANSION

With the IAM Enhanced File Structure, IAM files can acquire additional DASD extents when the file is being updated, in addition to the prior capability to take extents during file loads. This will eliminate many of the usability problems encountered with Compatible Format files due to their fixed overflow area size, and also eliminates the maximum overflow area size of 64,000 blocks. If the IAM file is a DFSMS Extended Format sequential dataset, then it can have up to 123 extents per volume. If the IAM file is a standard non-VSAM dataset, then it can have up to 16 extents per volume, and is limited by DFSMSdfp to a maximum of 65,535 tracks per volume.

MAXIMUM FILE SIZE

The maximum size of an IAM file is determined by a set of limitations imposed by DFSMS, MVS, and the architecture of the DASD devices. IAM's file size limitation based on the IBM 3390-27 DASD architecture is approximately 1,593 gigabytes of compressed user data for a DFSMS Extended format IAM dataset. For a standard non-VSAM format file, the limit is 201 gigabytes of compressed user data. These figures are based on the following calculations. For DFSMS Extended Format, using the maximum of 27 gigabytes per 3390-27 volume times a maximum of 59 volumes is 1,593 gigabytes.

The limit for an IAM file in the standard non-VSAM file structure as imposed by OS/390, or z/OS, is based on the DEB control block. This is a limit of 255 extents, with a maximum extent size of 65,535 tracks, which equates to using a maximum of 4,369 cylinders on a 3390. The maximum extent size is also the largest amount of space that can be allocated to a non-VSAM dataset per DASD volume. IAM datasets are further limited by the MVS TIOT structure to a maximum of 59 volumes. The maximum data storage for an IAM file can be obtained by using 1/2 track blocking, which is a block size 27,998 bytes, times 30 blocks per cylinder, which yields 820 KB per cylinder. Multiply the 820 KB per cylinder times 4,369 cylinders yields 3,499 MB maximum per 3390 device. (**Note:** this does not use the full capacity of the device.) The final maximum capacity of an IAM file is then calculated by multiplying 3,499 MB times 59 volumes, which yields 201 gigabytes. For 3390-3 type of DASD volumes, the size limitation is 154 gigabytes, and for all 3390-2 DASD volumes, the limitation is 102 gigabytes.

The above size limitation applies to IAM KSDS file types, and to IAM ESDS file types defined with the PSEUDORBA or XESDS attribute. For applications that require a true 4-byte VSAM RBA value for an ESDS file, such as the SAP product, the size limitation is 4 gigabytes of user data.

02.10 IAM DATA COMPRESSION

ADVANTAGES OF IAM DATA COMPRESSION

IAM can compress the data in both KSDS and ESDS files, and IAM is completely transparent to the programs that create and use those files. With IAM Version 8.0, customers can choose to use either standard IAM software compression, or they can request that IAM use the IBM hardware compression instruction. Use of the IBM hardware compression instruction is not recommended for customers that do not have at least an IBM zSeries processor, which offers CPU time reductions for hardware compression compared to earlier processors.

For the IAM software compression, Innovation selected a proprietary algorithm that is optimized for minimal CPU processing requirements that also provides good space savings for most files. IAM's software data compression does not rely on a compression table, which is one less exposure to potential data loss. While it is possible for alternative data compression algorithms to achieve a greater amount of record size reduction, the CPU time to achieve such results can be excessive. **In fact, for many files, IAM's software compression CPU time is still less than VSAM's CPU time without data compression!** Also, because of VSAM's use of generic compression tables, IAM still, for many files, is able to achieve similar or better space savings than VSAM hardware compression.

Starting with the zSeries line of processors, IBM has made dramatic improvements in the CPU time required for their hardware compression instruction. With this performance improvement, IAM Version 8.0 can now use the IBM hardware compression function. IAM's CPU time with hardware compression will be less than VSAM CPU time with hardware compression, and with customized compression dictionaries, IAM may achieve greater compression. Unlike the IAM software compression, the IBM hardware compression requires the use of a compression / decompression dictionary. With the initial support of the IBM hardware compression, IAM is providing a generic compression dictionary that is oriented towards English text, with some numeric data. This dictionary will provide for decent hardware compression for files with that type of content, including datasets that contain name and address data. Users can provide their own compression dictionaries based on the data that is actually in the dataset that is being compressed. Instructions on creating customized compression dictionaries are in section 10.61 of the IAM manual. Additional compression dictionaries and automated dictionary selection in the future may be offered by IAM in the future, if customers find hardware compression beneficial.

Data compression not only provides disk space saving it also reduces overall processing time. The benefits of using data compression with IAM include reducing the amount of data transferred over the I/O subsystem without a significant increase in CPU time.

RVA COMPRESSION

DASD devices that provide data compression, such as RVA devices, can be used with IAM files. However, there generally will not be additional space savings by using both compression techniques. Innovation recommends that IAM compression be used because of the advantages of reduced I/O transfer time, reduced physical I/O's, and reduced virtual storage which are only possible with an access method data compression technique.

02.10 CONTINUED . . .

ELIGIBILITY FOR DATA COMPRESSION

IAM considers KSDS type of files that are 75 tracks or larger, and all ESDS files as candidates for data compression. Setting the IAM DATACOMPRESS Global Option can easily change the file size that determines eligibility for IAM Data Compression. The automatic setting of IAM's Data Compression can always be altered through the use of the IAM Override facility.

To be eligible for IAM Data Compression, a file must be defined with a maximum record size that is at least 10 bytes more than its key length plus the relative key position (RKP). If Data Compression is enabled for a file, IAM will only compress individual records when the data following the record key exceeds ten (10) bytes in length. If compression would make a record larger than the original, IAM leaves the record uncompressed. Subsequent updates to an uncompressed record will keep the record uncompressed.

Data Compression can be used on all your IAM files. If a particular file is found by IAM to be uncompressible, there is no penalty in CPU time to process that file after the load. It is as if compression had never been requested for that file. There may also be a few files that just do not show much of a benefit from data compression. For example, SMP/E CSI files have an average record length that is just a bit larger than their key. When there is not much data to work with, there is little data compression can do to reduce a file's size. In these cases IAM's Data Compression may show little saving, beyond the space reduction that comes with simply converting to IAM. If a specific file shows only marginal compression there will likewise be only a marginal increase in IAM's CPU time to process that file.

A report on the estimated disk space saving that a conversion of your VSAM clusters to IAM file's with software Data Compression is available using the IAM VSAM Space Savings Analysis program IAMSIMVS. (See Section 42 - IAM Space Savings Analysis Program).

BACKUP COMPRESSED DATA

With the IAM Enhanced File Format, IAM offers the capability to backup and reload compressed data within an IAM file without decompressing or compressing the data. For large files, this is anticipated to allow IAM files to be backed up and reorganized faster than can be done today. Even when the data is compressed by the tape control unit, there is still the overhead of transferring all that data to the controller. With this new feature, both the CPU overhead and that I/O overhead is eliminated. The FDRREORG product from Innovation will automatically use this IAM feature.

The backup and reload of compressed data is specified for other programs, such as IDCAMS, by the use of the IAM Override facility. The override will have to be specified on both the backup and reload process, because IAM needs to know to not decompress the data on the backup side, and that the data is already compressed on the input side. Simply specifying the keyword BACKUPCOMPRESSED on the ACCESS and CREATE IAM overrides does the job. IAM adds four bytes to each record when performing this function, so any output file created will have to contain either variable (RECFM=VB) or undefined (RECFM=U) type of file. For variable output files, the record length for the output file (LRECL) will need to be at least 8 bytes more than the defined maximum record length for the file. For example, if the maximum record length for the file is 100, then the output LRECL must be at least 108. For undefined type of records, the maximum LRECL is 104, only 4 bytes more than the file maximum record size. Innovation recommends using RECFM=VB type of output to provide the best output device utilization.

Data that is in an IAM data compressed format on tape can be easily converted to an uncompressed format. Either reload the data with BACKUPCOMPRESSED into an IAM file, or use the IAMRECVR DECOMPRESS command to make a sequential copy of the dataset with uncompressed data. Note that you will need to know the original key length and key offset (RKP) to perform the DECOMPRESS function. IAM also provides a callable interface to read and perform the decompression from a data compressed sequential dataset that can be used by application programs.

For examples of using the BACKUPCOMPRESSED feature and using FDRREORG to reorganize IAM datasets, refer to Section 10.81.

02.20 IAM'S REAL TIME TUNING

With the introduction of the IAM VSAM Interface over fourteen years ago, IAM introduced a unique buffer management capability called Real Time Tuning. Real Time Tuning is a concept that IAM uses to dynamically manage the buffering for IAM files in response to the processing requirements. This capability allows IAM to adjust the number of buffers being used for any particular file, along with dynamically selecting appropriate buffer management and I/O techniques. This innovative and unique buffering technique has proven itself to provide extraordinary improvements in elapsed time to process batch jobs, as well as reductions in online transaction response times. IAM Real Time Tuning works within a range called MINBUFNO and MAXBUFNO values for the number of buffers, which can optionally be specified on a file-by-file basis by the user.

IAM buffering during a file load does not utilize the Real Time Tuning concept, because the file load I/O process is a sequential output process. The buffering technique used by IAM for a file load is described later in this section.

BENEFITS OF REAL TIME TUNING

The benefit of Real Time Tuning is the ability to easily achieve a level of performance that is difficult or impossible to reach with VSAM, along with a significant reduction in the manual effort and data gathering required for tuning your indexed files to meet the processing requirements. IAM's Real Time Tuning eliminates the need for deciding which buffering technique to use. With VSAM, you have to choose between NSR and LSR buffering. IAM has eliminated the need for index buffers, because IAM automatically keeps the index in virtual storage. For most datasets and applications, the default buffering provides for a very high level of performance. The IAMINFO run time report typically provides more than sufficient information to make any manual buffering adjustments that may be desired. This report, which is produced whenever an IAM dataset is closed provided that there is an IAMINFO DD card allocated to the job step, will even indicate with the IAM368 message if more buffers would have improved performance. The IAMINFO report can also be obtained from the IAM SMF records, if they are being collected, with program IAMSMF. In fact, that program can be used to print just the IAMINFO reports for those datasets for which IAM indicated that more buffers would have improved performance. This can be done by using the IAMINFO command with the ATTRIBUTE=MOREBUFFER keyword.

VSAM System Managed Buffering effectively decides for you whether to use NSR or LSR buffering, and will use more buffers than the normal VSAM default. Even with System Managed Buffering, you may still have to indicate through JCL what buffering technique will work best with the way the particular application uses the files. While jobs using VSAM files with System Managed Buffering may perform better than without, System Managed Buffering does not offer the dynamic buffer management that IAM Real Time Tuning offers. In many circumstances, IAM’s Real Time Tuning still does noticeably perform better than VSAM with System Managed Buffering.

02.20 CONTINUED . . .

REAL TIME TUNING FOR ENHANCED FORMAT FILES

The Real Time Tuning capability was significantly improved with the IAM Enhanced File format, which included:

1. The capability to manage larger number of buffers per file than with Compatible format. As shipped, IAM can have up to 2048 buffers per Enhanced format file. This capability can be easily increased if necessary. The limit for Compatible format files is 32 buffers.
2. The capability to handle concurrent I/O requests, both at the logical level and at the physical I/O level. This improves IAM responsiveness for online systems, for multi-volume datasets, and for volumes with Parallel Access Volume (PAV) capabilities.
3. Incorporate elapsed time into buffer adjustment algorithms. This will prevent the slow increase in buffers to lightly used files in an online system that is opened for several hours.
4. Provide sequential I/O read ahead, with I/O and processing overlap capabilities for synchronous mode (batch jobs) I/O processing. Up to a full cylinder's worth of data can be read in one physical I/O.
5. For mass sequential updates, IAM will write out multiple updated blocks per physical I/O.
6. IAM buffers are always requested to be in storage above the 16-megabyte line. However, should a buffer end up in below the line storage, IAM will freemain that buffer, to help avoid a potentially critical virtual storage shortage, unless that is the only buffer that IAM has available.
7. For Share Option 1 or 2 files, IAM also has incorporated empty block detection, which will prevent subsequent physical I/O for such empty blocks. This eliminates a problem encountered with rereading empty blocks when there had been large quantities of records deleted from the file.
8. The IAM I/O driver will also take advantage of the IBM Record Level Caching, for random I/O's and indicate sequential processing on the devices that provide those capabilities.
9. The IAM channel programs completely support the FICON and ESCON channels, utilizing the ECKD channel commands, for files on control units that support that capability. IAM continues to provide optimal use of the latest hardware in the market, by using block sizes that fully utilize the device's capability, and by using the latest features and capabilities when performing I/O.

BUFFERING FOR RANDOM I/O

For random processing, IAM dynamically acquires additional buffers up to the MAXBUFNO value, so that the most frequently referenced blocks can be retained in memory. IAM continually monitors what the added benefit would be for up to 32 additional buffers at any point in time by tracking contents and frequency of use. By monitoring requests in this way IAM can recognize patterns. After every 128 physical I/O's, IAM will evaluate the buffering. When IAM finds that a sufficient number of requests could have been satisfied without I/O to the disk within the past few minutes if more buffers had been available, an additional buffer is acquired. This continues until the maximum number of buffers allowed is reached.

IAM's buffer management techniques also provide for the release of buffers when it determines they are not providing any benefit in terms of reducing I/O within the past several minutes.

02.20 CONTINUED . . .

BUFFERING FOR SEQUENTIAL I/O

Buffer management for sequential processing differs from random in that IAM attempts to determine how many blocks it should be reading ahead in anticipation, to efficiently service the user. The goal is to be able to read as many useful blocks as possible with each real disk I/O. To accomplish this IAM monitors the number of blocks that are read consecutively, increasing or decreasing the number of consecutively chained buffers as appropriate. More buffers are acquired as necessary, at the rate of at most one per physical I/O, up to the MAXBUFNO value. Once processing has been completed for each block read sequentially, rather than aging the block out of the buffer pool, it is immediately treated as being the buffer with the least recent reference, because it is not expected that the block will be needed in the near future. For synchronous sequential I/O, which is typically done by batch jobs, when reading multiple blocks per physical I/O, IAM will make the data from each block available to the application program as soon as it is in storage. On mass sequential updates, when multiple consecutive data blocks have been updated, IAM will write out multiple blocks per physical I/O. This process is done asynchronously whenever possible, to allow for I/O and processing overlap.

The advantage of performing multiple blocks per I/O is improved performance due to a reduction in resource utilization. The primary savings are at the device level. This type of I/O will eliminate DASD seek and search time, as well as rotational positioning time because the device is already positioned to process the next block. This device time can frequently be well over 50% of the total device time needed when performing a single block I/O. Also eliminated is the CPU overhead involved with issuing an EXCP, scheduling the I/O, plus the CPU time involved with the I/O completion. Also, for batch jobs, IAM will attempt whenever possible to allow the sequential I/O to be active while allowing continued processing of the data that is already residing in IAM's data buffers.

MIX OF RANDOM AND SEQUENTIAL

Multiple random and / or sequential I/O operations can occur concurrently against any file. The use of the buffers will be balanced between the different requests based on the buffering needs of each request. The request with the highest buffer requirements will tend to utilize the most buffers. IAM does not lock out a request from a data block if another request is also using the block, such as VSAM does with the CI level lockout. A request may be temporarily delayed access to a block of data only if there is an actual physical I/O occurring, or during the actual processing time to move a record into or out of the buffer. This delay is managed internally by IAM, and does not result in deadly lockouts. The CI lock out problem caused by VSAM resulted in many installations not being able to use VSAM LSR buffering for some files under CICS, or forced files to small CI sizes to minimize the occurrence of the lock out problem. IAM eliminates these VSAM problems. IAM does perform the necessary record level lockout for concurrent updates to the same record to maintain data integrity.

SYNCHRONOUS VS. ASYNCHRONOUS I/O

For IAM datasets defined with Share Option 1 or 2 that are opened for update, IAM does distinguish the differences between synchronous and asynchronous processing. Synchronous processing is indicated by the RPL OPTCD=SYN, and is typically used by batch jobs. Asynchronous processing is indicated by the RPL OPTCD=ASY or OPTCD=WAITX, which is typically used by online systems, such as CICS. To improve batch performance IAM defers the rewrite to disk of a randomly updated buffer until the block residing in the buffer is forced out of the buffer pool. This technique reduces I/O whenever multiple rewrites are requested against data blocks in the buffer pool. Under Online systems (e.g. CICS), to insure integrity after a random update, buffers are always immediately rewritten. For sequential processing, the updates are deferred regardless of processing mode until IAM has several blocks to chain write, until the caller issues an ENDREQ request, or until the buffer is needed for another data block.

With sequential synchronous I/O, IAM is very aggressive with increasing the number of blocks that are being read per I/O, as well as being very quick to increase the number of buffers up to MAXBUNO. This is done to improve elapsed time for batch jobs. When asynchronous type of I/O is being done, IAM is less aggressive with sequential chaining. This is done to prevent tying up too many buffers at one time for online systems. This will prevent issuing an I/O request, which will monopolize a device for a relatively long period by reading many blocks per I/O, which could impact the responsiveness of the online system for other transactions.

02.20 CONTINUED . . .

SETTING MINBUFNO AND MAXBUFNO

IAM's Real Time Tuning works within an optionally specified range for determining the number of buffers called MINBUFNO and MAXBUFNO. MINBUFNO defaults to 1, unless explicitly overridden on an IAM Override statement. The value for MAXBUFNO can come from a number of sources, as indicated below in precedence order:

1. From the IAM ACCESS override value specified for MAXBUFNO.
2. From the IAM CREATE override value specified for MAXBUFNO when the file was loaded or defined.
3. The buffer space value specified on the IDCAMS DEFINE of the dataset, providing that it is greater than the IAM Global Options BUFSP value.
4. The higher of BUFND, BUFSP, or STRNO specified in either the JCL AMP parameter or explicitly in the ACB, provided that it is larger than the default MAXBUFNO determined from the IAM Global Options Table. (Note that JCL specification of one of these values takes precedence over the value specified in the ACB control block itself.)
5. A default value from the IAM Global Options Table, which will be the higher of either the MAXBUFNO Global Option, or the BUFSP Global Option (CICS will use CICSBUFSP) value divided by the file's block size.

For a typical 1/4 track blocked IAM dataset on a 3390 device, using the default Global Option values, IAM will set the MAXBUFNO value to 65 for batch, and 19 for CICS.

When an IAM dataset is opened, IAM will determine the values that it will use for MINBUFNO and MAXBUFNO as described above. It will then decide on a starting value. If MINBUFNO has been overridden, then IAM will initially acquire that number of buffers. If no MINBUFNO value has been specified for Enhanced format IAM files, IAM will initially acquire the higher of one half of the maximum number of buffers or the IAM Global Option value specified by BUFOPNO, which defaults to 4. If either the MINBUFNO value or the BUFOPNO value exceeds MAXBUFNO, they will be reduced to the MAXBUFNO value being used. The number of buffers that are in existence when the open processing completes may be higher than the original value. This is particularly true for files with a large index or large overflow area, both of which are read by IAM during open processing.

As a general rule, there should be very little necessity to specify a MINBUFNO value. The only reason for doing so would be to force IAM open processing to start at a higher number of buffers then the default values, which may speed up OPEN processing for extraordinarily large datasets. MAXBUFNO can be changed as desired, and generally should be increased when IAMINFO indicates that more buffers will be helpful, and you need to improve the performance for the dataset.

02.20 CONTINUED . . .

STATISTICS FROM IAMINFO

The IAMINFO report, which is produced when an IAM file is closed, will provide a variety of statistics relating to IAM Real Time Tuning and it's effectiveness. These fields include the following:

DISK BLOCKS READ—Indicates the number of physical I/Os (EXCPs) that were done to read data into the IAM buffer pool.

DISK BLOCKS WRITTEN—Indicates the number of physical I/Os (EXCPs) that were done to write data out to the file.

The TOTAL EXCP's issued by IAM for this dataset can be calculated by adding the DISK BLOCKS READ plus the DISK BLOCKS WRITTEN. That number will be equal to the EXCP value reported by SMF for this dataset, except for a file load with a temporary work dataset. For a file load with a temporary work dataset, the IAMINFO values are inflated by the I/O's done to the temporary dataset.

DYNAMIC BUFFER RETRIEVALS—Displays the number of I/O requests satisfied from the IAM buffers without performing a physical I/O. This statistic essentially identifies the savings of I/O due to IAM's Real Time Tuning.

MAXIMUM BUFFERS USED—Indicates the maximum number of buffers that were used to process this file.

MINIMUM BUFFERS USED—Indicates the MINBUFNO value in effect for this file.

MAXIMUM BUFFERS AVAILABLE—Indicates the MAXBUFNO value in effect for this file.

SEQ CHAINED BLOCKS READ—Indicates the number of data blocks that were read in as part of a sequential multiple blocks per I/O. These blocks were read as part of IAM's Real Time Tuning anticipation that the application program will use the blocks. This value plus the DISK BLOCKS READ value indicates the total number of blocks physically read into storage for this dataset.

SEQ CHAINED BLOCKS WRITTEN—Indicates the number of data blocks that were written as part of a sequential multiple block output I/O. This indicates the savings of physical I/O (EXCPs) due to IAM's Real Time Tuning accumulation of updated data blocks. This value plus the DISK BLOCKS WRITTEN value equals the total number of blocks written out to the dataset.

DO I NEED MORE BUFFERS?

While all of the above information is useful in evaluating the benefits of IAM's Real Time Tuning, there may still be a question as to whether or not more buffers would have been useful. The IAMINFO report will come out and tell you via an IAM attention message, message id IAM368 that is displayed in the run time INFO report. A flag is also stored in the IAM SMF record.

The IAM368 message is just an indication that the I/O performance for the specified file could potentially be improved by providing a larger MAXBUFNO value. It may very well be that the performance level of the job processing this dataset is quite acceptable, and if so there is no need to increase buffering. However, if you are looking for ways to improve the performance, then this message is an indicator of one of the possibilities for improving performance.

If the IAM368 message does not appear, then providing more buffers is very unlikely to change the performance.

For more information on tuning and getting the most out of IAM, refer to the Tuning Section in the IAM User's Guide portion of the manual.

02.20 CONTINUED . . .

FILE LOAD BUFFERING

As indicated at the top of the Real Time Tuning section, a file load does not make use of the Real Time Tuning concept. This is because a file load is exclusively a sequential output process. A fixed number of buffers are acquired at open time, based on the value for the CRBUFOPT Global Option or IAM Override. The required channel programs and I/O control blocks are built and initialized during open based on the quantity of buffers acquired. By default, for files that are allocated on cylinder boundaries, IAM will use enough buffer space to hold two cylinders, and write out a full cylinder per physical I/O. The default for track allocated files is buffer space for two tracks, and writing out one track per I/O. This buffering concept provides for I/O overlap, even for programs using synchronous processing. The EXCP counts will reflect the actual EXCP (physical I/O) operations done not the block count.

I/O processing during file load is different for DFSMS Extended format datasets. While the number of buffers will be the same as for EXCP processing described above, IAM is using BSAM access method to write out the data to the Extended format datasets. IAM will allow BSAM to write out multiple blocks per I/O. However, with BSAM output, it is not possible to control how many blocks will be written out per physical I/O. Also, because BSAM is being used, the EXCP count for file loads will be the actual number of blocks written out to the dataset, not the number of physical I/O requests. So, even though multiple blocks are being written out per I/O, that savings is not reflected in the EXCP counts.

02.30 AUTOMATIC DASD SPACE FUNCTIONS

When IAM files are defined with a secondary space value, IAM will automatically release over allocated space after the first load of an IAM file. This feature is provided for both Enhanced format and Compatible format files, with some differences in method of operation.

AUTOMATIC SPACE RELEASE

For Compatible format files, with their preallocated and preformatted overflow areas, upon completion of the file load process, the dataset is using as much DASD space as it will use, until the file is reloaded. Because this over allocated space is not likely to be used, IAM will automatically release that space. This feature is provided because IAM datasets typically require substantially less DASD space than VSAM, plus there is a tendency in many installations to over allocate datasets, particularly VSAM datasets.

RELEASE WITH SPACE RESERVE

The situation is a bit different with Enhanced format files, which have the ability to dynamically acquire additional DASD space as needed, and therefore have eliminated the allocation and formatting of blocks during the load process to accommodate file expansion. If IAM were to release all of the unused DASD space at the end of the file load, a dataset would very quickly end up taking extents as the file expands. Rather than eliminate all of the unused space, an automatic space reserve feature was developed. With this feature, some of the over allocated space will be retained rather than released at the end of the first load to allow for file growth without having to immediately go into secondary extents. The space reserved will be contained within the allocated extents after the file has been loaded. No additional extents will be obtained for the space reserve.

The desired amount of space to reserve will be calculated by using either the value specified for Overflow records on a CREATE IAM Override card, or by using the CA% Freespace value specified on the Define of the file. The calculation when CA% Freespace is used is similar to the calculation of the size of Overflow for the Compatible file format. Basically, the CA% Freespace value is cut in half, and then that percentage of the total DASD space required for the dataset as loaded will be set as the desired reserve quantity.

When the program loading the file issues the close, IAM looks at how much DASD space is allocated to the dataset, but is not yet used. If the allocated but unused space is less than the desired reserve quantity, then no space will be released. However, if the allocated but unused space is greater than the desired reserve quantity, then space will be released down to the desired reserve quantity. In all cases, the end of file is set at the end of the used area, and is indicated so in the VTOC LSTAR field. This will allow a DASD space management utility, such as Innovation's FDR/CPK product, to release the unused space that was reserved if it has not yet been used.

SPACE RESERVE AND RELEASE EXAMPLE

For example, consider a file where CA% Freespace is defined as 20%, and the file ends up using 300 tracks of space. A DASD reserve value of 30 tracks is calculated by taking half of the CA% Freespace, or 10% of the 300 tracks. Any allocated space over 330 tracks will be released. If the dataset is allocated to less than 330 tracks, then no space will be released. If the file was allocated initially with 600 tracks, then 270 tracks of space will be released, leaving a total of 330 tracks allocated to the IAM dataset.

If CA% Freespace is 0 and there is no Overflow override, then no space reserve will be done, and all excess allocated space will be released. In the example cited above, the dataset would end up with only 300 tracks allocated. If the desired space reserve quantity is less than one cylinder, then no space reserve will be done, because most files are cylinder allocated, and the space release is done at a cylinder boundary.

AUTOMATIC SPACE RELEASE SUMMARY

RELEASE is an especially important IAM feature. IAM files generally take 30 to 70% less disk space than VSAM. If the original VSAM IDCAMS DEFINE space allocation values were left unchanged without auto- RELEASE, a lot of disk space would continue to go to waste.

RELEASE is IAM's default and as most other IAM defaults it can be changed in the IAM Global Options table using the program IAMZAPOP (RELEASE= see Section 91).

02.30 CONTINUED . . .

DYNAMIC SECONDARY SPACE ADJUSTMENT

IAM will dynamically adjust the secondary space quantity for Enhanced Format IAM files that are in standard non-extended format. This function is provided because such datasets are limited to 16 extents per volume. The algorithm will take effect once a file has used five extents on a volume. At that point in time, IAM will increase the secondary space allocation by a multiplication factor specified in the Global Options Table, or from IAM Overrides. The default secondary space multiplication factor as IAM is shipped is ten for file loads, and five for file updates. The Secondary Space Adjustment feature is subject to the following rules:

1. The secondary space quantity will not be increased to value that exceeds the original primary space quantity.
2. The secondary space quantity will not be increased to a value that exceeds the size of the largest contiguous available extent on the volume. Note this is only effective once the file has obtained five extents.
3. If the original secondary space quantity is higher than the original primary space quantity, the secondary space quantity will not be adjusted.
4. For single volume files, the original secondary may be decreased from the original value to the largest extent available on the volume, just to try to keep it running as long as possible.

For example, if a file is Defined with a primary quantity of 500 cylinders, and a secondary of 10 cylinders, after five extents have been acquired, the secondary during a file load will be adjusted up to 100 cylinders. During a file load, the maximum space that will be used for this file is:

- 1 extent of 500 cylinders
- 4 extents of 10 cylinders each
- 11 extents of 100 cylinders

This will bring the total maximum space up to 1,640 cylinders, which is slightly less than the VSAM maximum of 1,720 cylinders. Although IAM allocated less DASD space than VSAM, the amount of user data kept in the IAM file will generally be larger due to IAM's space savings capabilities. This is due to IAM’s efficient use of DASD devices and Data Compression feature. So, by providing the Dynamic Secondary Space Adjustment feature, IAM files have the potential to grow in size as large as VSAM files will, although it will be done in fewer extents.

For this same allocation, the secondary will be adjusted up to 50 cylinders during a file update run. However, if the primary space is 20 cylinders, and the secondary is 10 cylinders, then the maximum value that will be used for the secondary is limited to 20 cylinders. The maximum secondary space quantity that will be requested for any particular file is included on the IAMPRINT LISTCAT output.

DYNAMIC SECONDARY SPACE ADJUSTMENT OVERRIDES

The secondary extent multiplication factor can be changed by IAM Overrides on a file-by-file basis, using the MAXSECONDARY keyword. Regardless of the value set, the basic rules for modifying the secondary space quantity remain as explained above. The factor can be specified on the IAM CREATE override during the file definition or file load, and on file updates with an IAM ACCESS override. The values permitted are from zero to ten. Values of zero or one will prevent IAM from increasing secondary allocation. The secondary allocation value may be reduced for single volume files when there is insufficient space for the secondary. When the secondary factor is specified on a CREATE override during file definition, the value is stored with the file. That value will be used for subsequent file loading and file updates, unless overridden at run time.

02.30 CONTINUED . . .

MULTI-VOLUME CONSIDERATIONS

With the IAM Dynamic Secondary Space Adjustment feature, there is an additional option for Enhanced format multivolume files that are defined without guaranteed space. Again, this function does not apply to DFSMS Extended Format IAM datasets. When it appears to IAM that the EOV request will result in the next volume being allocated, then IAM will request the primary space value, rather than the normal secondary quantity that is used for non-VSAM files. This feature can be controlled by either IAM Overrides, or by the IAM Global Options table. This feature is provided to offer an alternative so that IAM space allocations will be similar to VSAM. The IAM Override keyword is:

1. MULTIVOLUME=PRIMARY or
2. MULTIVOLUME=SECONDARY

These IAM Override keywords can be used on the CREATE Override statement. When specified on the CREATE keyword during file definition, the value specified is saved with the file control information and will remain the applicable option, unless overridden by a particular job step.

The rules for IAM files defined on DFSMS managed volumes with the Guaranteed Space attribute are different. This is because DFSMS will automatically allocate the primary space quantity on each volume when the file is defined. The secondary allocation quantity will be modified as described above for single volume files.

Another special multivolume circumstance is when a file is defined with no secondary space specified. For most circumstances, IAM will set the secondary allocation value to the primary value. For datasets on DFSMS managed volumes with Guaranteed Space, the secondary is left as zero, with the primary being allocated on each volume when the file is defined. For systems where DFSMS is active, but the datasets are on non-SMS managed volumes, IAM will leave the secondary as zero, and allocate the primary space on each volume when the file is defined. This is to mimic the DFSMS Guaranteed Space, and will prevent any secondary extents from being taken.

02.40 IAM ALTERNATE INDEX SUPPORT

ALTERNATE INDEX SUPPORT

IAM offers, at an additional cost, Alternate Index Support. With this optional feature, Enhanced Format IAM datasets (both KSDS and ESDS) can have associated alternate index datasets, which will also be IAM datasets. All of the features and benefits of IAM are available to base clusters and all of their related alternate indexes. There are no programming changes necessary to use the IAM Alternate Index support. No changes are required to CICS. Simply change the define of the base cluster to include either OWNER($IAM) or include $IAM in the base cluster name, or use a DATACLAS or STORCLAS with $IAM in the class name. Subsequent define steps for any related alternate index datasets or paths will be automatically converted to IAM, when the base cluster is an IAM dataset. The alternate index support is designed to provide the same high level of compatibility with VSAM alternate indexes that IAM has been providing for KSDS and ESDS types of datasets. Some dataset management utility jobs for backing up, restoring, and renaming datasets involved in an alternate index relationship may need to be revised.

THE IAM ALTERNATE INDEX

An alternate index provides an additional index to an indexed type of dataset (KSDS), or an index to an entry sequence dataset (ESDS). Users can define one or more alternate indexes to any base dataset. The alternate index dataset itself is an IAM KSDS enhanced format type of dataset that is indexed by the alternate key. The records in the alternate index contain some control information, the alternate key, and the primary key value for the record in the base dataset with the corresponding alternate key. An alternate index is defined as containing UNIQUE keys when there is only one base record with any specific alternate key. Alternate indexes can also be defined as containing NONUNIQUE keys, where any particular alternate key can be contained in multiple base records. As with VSAM, the alternate index dataset can be explicitly processed by programs without referencing the base cluster, or can be used to access the records in the base cluster. To use an alternate index dataset to access the records in the base cluster, a PATH must be defined.

An additional attribute of alternate index datasets is whether they are upgradeable. When an alternate index is defined with the UPGRADE attribute, IAM will automatically update the alternate index either when updates are made to the base cluster, through the primary key or when accessed through an alternate key. Any alternate index defined with the NOUPGRADE attribute will not be automatically updated by IAM, and it is the application programs responsibility to ensure that the alternate index is updated in a manner to remain synchronized with the base dataset.

THE IAM PATH

A Path provides the mechanism to access a base cluster through an alternate index. When a path is defined, one must provide the name of the path and the name of the related alternate index to be used to access the base cluster, or the base cluster to be processed whenever the path itself is referenced. The UPDATE or NOUPDATE attribute is specified for each path defined. When a path is defined with the NOUPDATE attribute, any upgradeable alternate indexes will not be updated automatically when this path is used to access a base cluster either directly or through an alternate index.

For VSAM datasets, a PATH is only a catalog entry that is quite similar to an ALIAS entry. With IAM, a PATH will become a one-track dataset containing the name of the related alternate index or base cluster and the UPDATE or NOUPDATE attribute of the PATH. A single track dataset was chosen instead of an ALIAS type of catalog entry because of concerns about there not being adequate dataset management utility program support for ALIAS entries, which might result in the loss of the entry.

02.40 CONTINUED . . .

USING AN IAM ALTERNATE INDEX

The procedure for defining and using IAM datasets with alternate indexes is identical to the VSAM process. The only difference is to specify on the DEFINE that the base cluster is to be an IAM dataset. This specification is typically done by either adding the OWNER($IAM) parameter to the define parameters, changing the name to include the $IAM literal, or using a DATACLAS or STORCLAS with $IAM in the class name. Most applications that are going to be using IAM for their alternate index already have the general procedures established. To convert to IAM, the only change needed is on the define step of the base cluster. An overview of the process for using IAM alternate indexes is described below. All of these steps are covered in detail in the User's Guide (Section 10) of this manual.

- Defining the base cluster.
- Loading the base cluster with data.
- Defining the alternate index(es)
- Building the alternate index(es), usually with IDCAMS BLDINDEX command.
- Defining paths to the alternate index(es) and if used, paths to the base cluster.

In general, there are no JCL changes required when using IAM Alternate Indexes and Paths. To access a base cluster through an alternate index, simply specify the name of the Path on the DD statement, just as is done with VSAM. When the Path is opened, IAM will dynamically allocate and open all of the required alternate indexes and the associated base cluster using the relation information stored within the IAM file structure. When the Path is closed, IAM will close and de-allocate all of the alternate indexes and the base cluster that were allocated during the open process.

Because the IAM alternate indexes and paths are non-VSAM datasets, they will not be automatically grouped together in a sphere with their associated base cluster, as is typically done by many dataset management utility programs. Some changes to your dataset management procedures may be necessary when using IAM Alternate Indexes. After renaming a component of an IAM sphere, or performing a restore or copy to a new name, a define recatalog will be necessary.

02.50 IAM'S DYNAMIC TABLING: 'DATA RECORDS-IN-VIRTUAL'

IAM has another feature that can potentially reduce physical I/O's for files that are randomly read. This feature, called IAM's Dynamic Tabling, offers significant performance benefits for some applications. With this feature activated, IAM tables records retrieved randomly from a file in virtual storage, without any programming changes to existing applications. Then, on subsequent random reads, IAM checks to see if the key requested is for a record contained in the virtual table. If the record exists in the table, IAM passes it back to the user, eliminating the I/O to the disk. If the record does not exist in the table, it will be moved into the table.

NOTE: Random reads which are eligible to use the dynamic tabling feature are identified in the IAMINFO report as R. (READ) commands for Compatible format files, and as GET RANDOM commands for Enhanced format files. Other types of retrievals, such as GET commands and Read-or-GetNext cannot use the dynamic table because the precise key being sought is unknown.

To enable this option, use the IAM ACCESS Override statement. The keyword on the control statement is DYNCORE= (See Section 30). The DYNCORE value is specified in 1024 byte (1K) increments. The following example will reserve 200K of storage for the Dynamic Table.

```
ACCESS DD=iamfile,DYNCORE=200
```

IAM's Dynamic Tabling of data records is similar to the Data-In-Virtual concept IBM introduced with Linear Data Sets, without requiring any programming or file format changes. IAM's concept of tabling is more efficient than IBM's because it's oriented to individual records instead of 4K sections. A smaller amount of memory is used requiring fewer real pages to back up the virtual pages.

If the key being requested is not currently in the table, IAM reads the record from the file. If found, the record is passed to the user and tabled for subsequent retrievals. If the record is updated, IAM changes the record in the table and on disk.

Applications which will benefit the most from Dynamic Tabling are those with high file activity where a subset of records in the file are repeatedly being read, with few ever updated. Small files with high random activity and few updates become in core tables without the need for any programming changes.

Under MVS/ESA and OS/390, Dynamic Table storage is in extended private.

IAM's run time INFO report reflects the way an application uses the file. The report includes statistics on requests processed, I/Os to disk, Dynamic Table usage and the number of records retrieved from the Dynamic Table.

DYNAMIC TABLE RETRIEVALS—displays the number of record requests satisfied from IAM's Dynamic Table.

DYNAMIC TABLE RECORDS—displays the maximum number of records in the table.

NOTE: IAM Dynamic Tabling facility will not be used for files that contain spanned records, due to the very large amount of storage that would be required.

02.60 IAM'S VSAM TRANSPARENCY

IAM's system level VSAM Interface provides application program transparency. IAM allows an unaltered application program executing under OS/390 or z/OS to access IAM files in place of VSAM KSDS or VSAM ESDS files. IAM can be used in conjunction with the common programming languages COBOL, BAL (assembler), PL/1, RPG, C and any higher-level language products, which support keyed access to VSAM files.

IAM supports programs executing in AMODE(31) and VSAM control blocks (ex: ACB, RPL) residing above the 16MB line.

IAM supports the full range of VSAM file access commands GET, PUT, INSERT, GETPREV, ERASE, POINT, etc. and the file status commands SHOWCB, TESTCB, GENCB, and the VSAM catalog lookup macro, SHOWCAT.

IAM supports the following functions of IDCAMS, as they relate to VSAM KSDS or ESDS file processing: BLDINDEX, DEFINE, DELETE, LISTCAT, PRINT, REPRO and VERIFY. DELETE, PRINT, REPRO and VERIFY provide the same services for IAM files as they would VSAM clusters.

IDCAMS DEFINE will create an IAM file whenever the OWNER($IAM) parameter is specified, '$IAM' is placed somewhere in the dataset name, or $IAM is part of the Data Class or Storage Class name.

LISTCAT ALL displays IAM files as non-VSAM in its standard SYSPRINT report. LISTCAT ALL also displays the file's IAM characteristics in an IAMPRINT DD report, which will be dynamically allocated if necessary.

IAM has full support for the SMS environment. This support includes recognizing and using the SMS classes for allocation, honoring file attributes specified in the Data Class, and support for JCL allocation of IAM files; including temporary dataset support. Simply place $IAM in the SMS Data Class or Storage Class name to define a file as IAM. When using the SMS JCL allocation feature specifying $IAM in the dataset name, in the Storage Class name or in the Data Class name on the DD Statement results in an IAM file being allocated.

Activating the IAM VSAM interface is a simple procedure. While evaluating IAM, you can activate and deactivate IAM at any time. To start IAM all you need to do is submit the procedure, 'VIFSTART', supplied in the IAM Installation Control Library. Within a few seconds, IAM will be active in the system. Once testing has been completed and IAM is in production, you can activate IAM automatically each time the system is IPL'd.

Section 90.20 of this manual documents activating the IAM VSAM Interface.

02.70 SMS SUPPORT IN IAM

OVERVIEW IAM provides support for SMS that is equivalent to the VSAM support, including support for JCL file definition, and temporary datasets. By definition, to be eligible for an SMS managed volume, the file must be assigned a Storage Class. The Storage Class, along with optionally a Data Class and/or Management Class, can be explicitly specified on the DEFINE command, by JCL parameters for JCL defined files, or selected by the ACS routines. IAM files on SMS managed volumes will be cataloged with the class names. As a part of the SMS support, an additional method of triggering an IAM DEFINE is available. Files will be defined as IAM files if the Data Class or Storage Class name contain the $IAM literal.

SMS AUTOMATIC CLASS SELECTION ROUTINES For both IDCAMS DEFINE's and VSAM JCL allocations, the ACS (Automatic Class Selection) routines are called prior to the IAM DEFINE intercept. When IAM intercepts the request, the SMS classes, the SMS Storage Group, and the SMS volumes have already been selected. IAM will then screen the request and determine if the file should use the IAM format. If $IAM, in the Data Class (DATACLAS) or Storage Class (STORCLAS) name, is being used as the criteria for determining IAM format files, then the class name(s) must have $IAM in them at this point in the process. They contain either the explicit names from the DEFINE request, or the name(s) selected by the ACS routines. This allows the installation the possibility of controlling IAM files, and IAM usage through the ACS routines. For JCL allocation, these are the classes and volume(s) that will be used.

For IDCAMS DEFINE requests, the ACS routines will be re-entered when IAM issues the dynamic allocation of the file as a non-VSAM dataset. The request will specify the SMS classes as received from the intercepted DEFINE request, and the volume(s) that had initially been selected by SMS. While Innovation does not recommend this, the ACS routines can change the SMS classes and the Storage Group, which will change the volume(s) on which the file is placed. The ACS routines can check the &DSORG value, which will be VS (VSAM) on the DEFINE, and PS on the IAM dynamic allocation. At this point, the file must not be switched to a non-SMS volume, because the allocation will fail. However, it can be switched from an unmanaged volume to an SMS managed volume. Changing the DATACLAS at this time will have no effect on the file characteristics, as they were determined by IAM prior to the dynamic allocation. The MGMTCLAS, STORCLAS, and Storage Group can all be effectively changed by the ACS routines on the dynamic allocation request.

SMS CLASSES IAM provides full support for IDCAMS defines under SMS. The DATACLAS, STORCLAS, and MGMTCLAS can either be explicitly provided on the DEFINE command, or selected by the ACS routines. The Data Class can provide file characteristics for the file being defined, including record length, key length, key offset, share options, free space, and others, eliminating the need to specify those values explicitly on the DEFINE. As per SMS rules, the options in the Data Class will be used, unless explicitly overridden on the DEFINE command.

DFSMS EXTENDED FORMAT In IAM Version 8.0, enhanced format IAM files may now be optionally defined as DFSMS extended format sequential datasets. These datasets are now able to fully utilize larger volumes, such as a 3390-9 or even larger volumes of up to 32K cylinders. These datasets can use up to 123 extents per volume and exceed the prior limit of 255 extents in total when multiple volumes are specified. Such datasets must be DFSMS managed, with a data class that specifies “Extended Required” or “Extended Preferred” for the dataset name type, and with Compaction set to NO. These datasets must also have a Storage Class with a “Sustained Data Rate (SDR) either left blank, or set to 0. Setting other values for Sustained Data Rate will cause the dataset to have multiple STRIPES, which IAM does not support. IAM will internally compress the file, with either software or hardware compression.

ALLOCATION ERRORS If the IAM allocation encounters any errors, the error messages will appear on the JES job log, with the MVS allocation messages (SYSMSGS) and also on the IDCAMS SYSPRINT, if it is available. Due to the manner in which IDCAMS prints messages on SYSPRINT, the error messages from IAM will precede the actual DEFINE command. IDCAMS will also print out additional error messages after the DEFINE, performing an analysis on the return codes set by IAM. Whenever possible, IAM uses the VSAM return codes that most clearly indicate the actual problem, although that is not always possible. Always refer to the IAM and related allocation error messages for the most precise problem determination possible.

02.70 CONTINUED . . .

JCL ALLOCATIONS VSAM files being defined through JCL can also be easily converted to IAM files. This is done by either putting $IAM in the dataset name (DSN), or by using a Storage Class (STORCLAS) or Data Class (DATACLAS) with $IAM in the name. Both permanent and temporary datasets can be defined, with the restriction that temporary datasets cannot be multivolume, same as with VSAM. The use of a Data Class (DATACLAS) is highly recommended for JCL defined files. By using a Data Class, values for Freespace, CI Size, and Share Options can be specified, which are not available through JCL parameters. All files defined in JCL will, by default, be capable of handling variable length records, up to the maximum length specified in the DATACLAS or LRECL field.

To allocate IAM files through JCL, IAM must be in the linklist. STEPLIB and JOBLIB are ineffective in this case, because it is the initiator that is issuing the allocation, and IAM must have access to various load modules for the define.

If the IAM allocation encounters any errors, the error messages will appear on the JES job log and with the MVS allocation messages (SYSMSGS). SMS will also print out additional error messages appearing with the MVS allocation messages, performing an analysis on the return codes set by IAM. Whenever possible, IAM uses the VSAM return codes that most clearly indicate the actual problem, although that is not always possible. Always refer to the IAM and related allocation error messages for the most precise problem determination possible.

Any CREATE overrides for JCL allocated files must be in the job step that loads the file, not necessarily the step allocating the file. The define process does not access the IAMOVRID DD for JCL defines. For example, if the file is allocated in an IEFBR14 step, and then subsequently loaded by an IDCAMS REPRO, the IAM create overrides must be in the IDCAMS REPRO step.

DYNAMIC ALLOCATION The TSO ALLOC command, and the MVS DYNALLOC service, has also been enhanced to provide for allocation of new and temporary VSAM files. These requests will also be screened by IAM, and can be converted to an IAM file in the same manner as a JCL allocation can. IAM treats the request just like a JCL request. The new ALLOC keywords are the same as the new JCL keywords, and dynamic allocation has the equivalent text units.

MULTIVOLUME IAM files can be spread across multiple SMS managed volumes, both with and without Guaranteed Space. Note that IBM restricts temporary VSAM files to a single volume, this also applies to IAM. When an IAM file is defined with a Storage Class that specifies Guaranteed Space, the primary allocation quantity is allocated on each volume at DEFINE time, as per the SMS non-VSAM rules. When Guaranteed Space is NOT specified, only the first volume is selected at DEFINE, and the subsequent volumes are cataloged as an '*'. During file load or reorganization, if additional volumes are needed, they will be selected by SMS. SMS has a restriction that within a job step, if a file defined without guaranteed space uses additional volumes, only one DD statement can be used, because any other DD's are not updated to indicate the additional volumes. For IAM users, this is only of concern for job steps loading or reorganizing files and being accessed within the same step.

02.80 OTHER FEATURES OF IAM

SPANNED RECORD SUPPORT

Enhanced format IAM files can contain spanned records. A spanned record is one that is larger than the physical block size of the IAM dataset, and will require multiple physical blocks to store it. With IAM, the record size limitation is the amount of data that IAM can fit within 256 data blocks. With a maximum block size of 32760, and IAM file can now have a maximum record length of approximately 8 megabytes. Spanned record support will allow IAM Alternate Index customers to have large numbers of records in a base cluster to have the same alternate key. The spanned record support also eliminates the dasd space waste that occurs on files where the record size requires a physical block size that exceeds the half-track block size that resulted in IAM using a block size of 32760.

Please be aware that many utilities used to process IAM (and VSAM) files do not support record sizes of approximately 32760 bytes or larger, including IDCAMS and the various SORT utilities. The IAM file recovery utility program, IAMRECVR, does provide support for these large records.

IAM INDEX DATA SPACE

The nature of this new index structure is that it will require more virtual storage than the old overflow index structure. To help minimize the increased storage requirements, IAM is now using a compressed key format for the overflow index, reducing the size of almost all of the entries in the overflow index. Additionally, IAM will automatically for CICS, and optionally for batch jobs, store the prime index and overflow index in a Data Space. IAM will create only one Index Data Space per job, sharing the Index Data Space storage for all open Enhanced Format IAM files. This Index Space will significantly reduce IAM's use of private address space virtual storage, relieving some of the virtual storage constraint problems particularly for CICS regions with hundreds of open IAM datasets. Data space usage will be monitored and reported by the IAMXMON transaction, and in the IAMINFO reports. To use the new IAM Index Data Space, customers must be running OS/390 or MVS/ESA 4.2.2 or higher.

ESDS EXTENDED ADDRESSABILITY

IAM has support for 8-byte RBA values in the IAM ESDS type of files. This is referred to by IBM as Extended Addressability. VSAM ESDS datasets defined with this DFSMS attribute can exceed 4 gigabytes by using an 8-byte RBA, instead of the normal 4-byte RBA. An IAM 8-byte RBA file is created by either specifying the XESDS keyword on the IAM CREATE override, or through specification in the DFSMS Data Class. Such IAM ESDS files do not have to reside on DFSMS managed volumes, as do VSAM ESDS files. Your application programs may not be able to take advantage of the 8-byte RBA values without coding changes. For such situations, you may find that the IAM PSEUDORBA feature, which provides for over 4 gigabyte ESDS files with a non-standard 4-byte RBA as an easy solution.

PSEUDORBA type ESDS files are created through the PSEUDORBA IAM CREATE Override. This enables many applications to exceed the 4-gigabyte size limitation without changing to an 8-byte RBA value. The limitation is that if an application is dependent on the RBA value being identical to the VSAM architecture, then PSEUDORBA cannot be used. The only known application package that has this limitation is SAP.

MASS SEQUENTIAL DELETES

For Enhanced Format files, IAM has incorporated an ability to temporarily logically delete a record from a data block. Then, when that data block is about to be written out to DASD, the records are physically deleted from the data block. This eliminates the overhead of constantly moving records within a data block, as prior records are deleted, only to end up being deleted as well. In the case where every single record is deleted from a data block, this enhancement eliminates the data movement entirely from that process.

A further difficulty has been rarely encountered after such mass deletes, where certain types of I/O requests result in the empty data blocks being read repeatedly, often unnecessarily. IAM does sequential processing without referring to any index structure, because the internal structure of the file does not require such overhead. IAM has no way to know that a prime data block is empty, without actually reading that data block. With IAM Enhanced Format files, that are defined with Share Option 1, or that are opened for update with Share Option 2, will now keep track of those empty blocks, and not reread such blocks in sequential modes of processing. This support is limited to particular circumstances, because in other situations there could be another ACB opened for UPDATE against the same file, and therefore there can be no presumption about the contents of a data block.

02.80 CONTINUED . . .

DYNAMIC REGION SIZE ADJUSTMENT

As customers have converted files to IAM, they occasionally hit the MVS default limit of 32 megabytes extended private storage or their REGION size limit. This has in the past necessitated the modification, or in many circumstances writing, and supplying an IEFUSI exit. Innovation has developed, and distributed a sample IEFUSI exit, that frequently can be used with minor modifications. This exit is distributed in the ICL (Installation Control Library) provided with IAM.

Because many customers have unexpectedly hit this limitation, IAM has the ability to dynamically increase the extended private region limit. When IAM is opening an Enhanced Format File, it estimates the amount of storage that will be required to open the file. Included in that estimate is storage for the prime and overflow indexes, control information, and buffers. IAM then checks to see if that amount of virtual storage is available. If not, IAM will then attempt to increase the extended private storage limit by the quantity required to open the file, rounded to 4 megabytes. By default, IAM will not increase the extended private region to greater than 512 megabytes. This maximum value can be either increased or decreased through the IAM Override facility, or by changing the IAM default in the IAM Global Options Table.

While processing a file, if a critical storage acquisition fails, that will also drive the dynamic region adjustment, with the same limitations as above. A storage acquisition is considered critical if it is required to successfully complete a request. For example, one critical storage acquisition is to expand the size of the overflow index to complete an insert or update request. If storage cannot be obtained, then the insert or update request will fail with a File Full VSAM logical error. With Dynamic Region Adjustment, when a critical getmain fails, IAM performs the region adjustment. IAM then retries the failing getmain, and if it fails again, then the request is failed. Dynamic Region Adjustment will not be invoked to increase buffers, because processing can continue without failing requests.

Dynamic region adjustment affects only normal file access; it does not function during a file load process. The value can be changed in the IAM Global Options Table, by using the keyword MAXREGION. Refer to Section 91 for information on changing IAM Global Options. The region adjustment can also be controlled with an IAM Override. The IAM ACCESS override keyword is MAXREGION, and it specifies the upper limit for the above the line storage, in megabytes. Note that by specifying the keyword DD=&ALLDD, the value will be effective for all Enhanced Format IAM files, unless there is a specific override.

MULTIPLE ACB SUPPORT

For Enhanced Format Files, IAM will recognize that an ACB has already been opened for a file, within the same task, and share the index structure and buffers. For applications, or CICS regions, where a file was opened under one DDNAME or ACB for read only, and the other for UPDATE, they occasionally experienced an inability to retrieve an updated data record. This will no longer be the case. This support also reduces the storage requirements for having multiple ACB's opened to the same file.

A special feature is available when the opened ACB is read only, and an ACB is subsequently being opened for update in the same address space. When that occurs, the overflow index is completely rebuilt, and all the buffers refreshed so that any updates that may have been done by batch jobs will be automatically and immediately available once the update ACB has completed the open process. For online systems that utilize two ACB's, where one is read only and the other update, the update ACB can be closed to allow batch updating. The read only ACB can remain open for processing, however it may not have access to all of the records updated by the batch job(s). When the batch updating terminates, the update ACB can be reopened, and access to all of the batch updates will be immediately available.

02.80 CONTINUED . . .

IAM JOURNALING

IAM provides an optional automatic journalling capability on file updates for Enhanced format IAM KSDS and ESDS types of files. This capability is provided to assist in improving data availability, particularly for the very large multivolume datasets. The improved availability of the data is accomplished by reducing the frequency of dataset backups, and providing a speedier recovery facility for failing batch jobs.

An example of the intended use is as follows. Let's say there is a multivolume IAM dataset that is updated every day, from both online and batch. Because of this, the dataset is backed up every day. With IAM journalling AFTER images, the file may now need to be backed up only once a week, however the smaller log dataset containing the updates will be backed up daily. If a recovery is needed, the dataset is first restored from the last backup, and subsequently all of the updates are reapplied from the IAM journal to the dataset using the IAM journal recovery program IAMJREST. The data availability is improved by elimination of the time spent backing up the dataset every day, which may take well over a couple of hours.

The other intended use is to provide a backout mechanism of updates performed by batch jobs. To utilize the backout capability, the user must indicate that the BEFORE images of updated records are to be included in the journal. Should a batch job abend, rather than restoring the file and rerunning prior batch updates, all that needs to be done is to backout the updates from the batch job step (or entire job, and other jobs). This is accomplished by using the IAM journal recovery program, IAMJREST.

The IAM journalling feature is activated through the JRNAD override keyword, with the appropriate value of BOTH, BEFORE or AFTER indicated for the types of journal records required. As stated above, the AFTER images are necessary for performing a recovery from a dataset that has been restored. BEFORE images are needed when backing out dataset updates. The BOTH keyword enables collection of both the BEFORE and AFTER images. When specified on the CREATE override during a file definition or load, IAM will automatically journal all updates made to the dataset. When specified during file updates on the ACCESS override card, the specified journal options are only applicable to that job step. Additionally, the user must allocate a sequential dataset to contain the journal records, with an adequate amount of space. The name of the dataset must be the same as the IAM dataset name (i.e. cluster name) appended with '.LOG'. If the dataset name is over 40 characters long already, and if there is a '.' in position 40, then the dataset name will be 43 characters long, ending with the '.LOG" string. If the dataset name is over 40 characters long, and position 40 is not a '.', then the log file name will be 44 characters long, ending with the '.LOG' appendix. The user is also responsible for the management of the IAM journal dataset(s). They must be backed up as necessary, and also emptied out at various points. The log dataset can be emptied by deleting and reallocating the dataset, or by using an IEBGENER to copy into the log dataset from a DD DUMMY.

IAM journalling is not intended as a substitute for CICS journalling and the CICS transaction backout capability. Those capabilities must still be utilized if required.

Full information and examples on using the IAM Journalling capability are provided in section 10.88 of the Users Guide.

THIS PAGE WAS INTENTIONALLY LEFT BLANK

03.01 IAM DATASET STRUCTURE OVERVIEW

FILE STRUCTURE OVERVIEW

IAM provides improved levels of performance, efficiency and reliability unsurpassed by any other index file processor. To gain these advantages, rather than attempting to manage a VSAM structure more expertly, IAM establishes its own uniquely structured dataset.

The organization of an IAM dataset is structurally simpler than its VSAM equivalent. An IAM dataset is a relative block non-VSAM (DSORG=PS) organized and managed by IAM using the EXCP access technique.

Complementing overflow structures within an IAM file can accommodate any type of file growth. A percentage of every block in an IAM file can be left free at load time to accommodate randomly distributed additions. IAM establishes free records within the file to be used when inserts are not randomly distributed but are clustered into groups. For Enhanced format files, free space can be reserved during file load to accommodate file expansion. For Compatible format files, free blocks can be established during file load to accommodate inserts to the file.

Programs that create IAM files are device independent. IAM automatically determines the target device type and, using the file's record length and CISIZE, calculates the best blocksize for that device. Programs that process IAM files are totally insulated from the blocking and structure of an IAM file.

IAM FILE ALLOCATION

An IAM file is allocated on the disk and cataloged using an IDCAMS DEFINE or functionally equivalent operation, such as VSAM JCL allocation, or various TSO ISPF panels offered by various vendors. The IAM file is created when the file is opened for OUTPUT and loaded with one or more records sequentially by key.

IAM files can be loaded into a new allocation or an existing file allocation with a disposition of OLD or SHR as appropriate. IAM files are always considered to be reusable, unless the IAM Global Option ENABLE=NOREUSE is set, and the file is defined with the NOREUS attribute. The exception to this is that once an alternate index is defined for a base cluster, the base cluster becomes not reusable to retain full VSAM compatibility and because there is no mechanism to update the alternate index during a file load.

IAM files are simple non-VSAM space allocations managed via the EXCP access technique. IAM datasets contain fixed length blocks, within which IAM manages the data, index, and file description information. IAM datasets can be defined across multiple volumes. They make full use of secondary space allocation and do not require contiguous (CONTIG) extents. Space for an IAM file can be defined in records, tracks or cylinders.

For Compatible format files, the amount of space requested should be sufficient to accommodate all of the records to be loaded and the overflow defined for the file. If a secondary allocation value is defined, during a load procedure, should the primary allocation be insufficient IAM will acquire additional extents. VSAM's 4.3 Gigabyte (GB) file size limitation on non-SMS VSAM files does not apply to IAM files.

03.01 CONTINUED . . .

IAM FILE INTEGRITY

IAM files have been designed to provide the highest level of both file and data integrity. Some of the design factors that enhance file integrity include a prime index structure which will never be updated, minimizing as much as possible the need to update other control information about the file structure, designing a structure of adding to the existing structure, as opposed to modifying the existing file structure. The dynamically extended portion of the file structure is validated during open, with efforts made to recover from most anticipated type of structure errors that may occur. Every effort has been made to provide an expandable, non-destructible file structure that provides an outstanding level of performance and responsiveness.

As with all prior versions of IAM, the access method will never pass back to an application program the address of an actual buffer. This is done to prevent inadvertent corruption of the data contained within the buffers, as much as possible. Even for requests indicating OPTCD=LOC (locate) mode processing, IAM will always make a copy of the record in another storage area, rather than pass back the address of the data record within the buffer. Also, with the IAM Enhanced File Format code, all of the critical data areas are acquired within their own 4K pages, to prevent the inadvertent storage overlays that sometimes occur when there are multiple programs using the same page of virtual storage.

In spite of all these efforts, should a problem occur within the IAM file structure, IAM provides a recovery program, IAMRECVR, which will recover all the data that it is possible to recover for the file. This program reads the data blocks on it's own, without going through the index. This program can be used for recovering from physical I/O errors, or structural errors that may have occurred.

SPECIFYING FILE STRUCTURE TYPE WITH OVERRIDES

For KSDS type of files, the user has the option to specify either the Enhanced file structure, or the Compatible file structure. File structure is determined when the file is defined, either with the IAM CREATE override, or defaulted based on the IAM Global Options Table. IAM is shipped with the IAM Global Options Table default set for the Enhanced File Structure. Existing files that are reorganized without being deleted and redefined will retain their previously existing type of file structure. Files that are deleted and defined will default to the file structure specified in the IAM Global Option table.

ESDS type of IAM files is always in the Enhanced File Format.

The file structure can be specified on the IAM CREATE Override control statements using one of the following keywords:

1. **ENHANCED** For Enhanced format files, or
2. **COMPATIBLE** For a format compatible with prior versions of IAM.

While the file structure type override can be specified on the file load, it is **highly recommended** that it be specified for the file definition. Some decisions are made during file definition that may result in certain capabilities not being fully functional. For example, if a file is defined as an Enhanced format file, but then changed to Compatible format, then no value for the size of the Independent Overflow area has been determined, which can lead to the file immediately becoming full. On the other side, if a file is defined as Compatible and has a space allocation specified in records, then if changed to Enhanced format on the file load, the Dynamic Secondary Space Adjustment feature may not function, due to how the space allocation is saved.

03.02 ADDING RECORS TO AN IAM FILE

VSAM CONCEPT

VSAM uses two concepts to manage inserts in a VSAM KSDS file:

FREESPACE CI%—Specifies the percentage of each control interval that is left empty for future inserts. IAM's Integrated Overflow is basically the same concept, and utilizes the value specified for CI% Free Space.

FREESPACE CA%—Specifies the percentage of CIs within each CA to be left empty. Its purpose is to establish free space areas throughout the file for CIs that are filled. These areas are tied to the file's existing index structure. If an insert does not fit within the CI where it should be placed, VSAM will split the CI, using one of the free CIs within the same CA. If a CA has no remaining free CIs, VSAM splits the CA into an area at the end of the file.

The VSAM technique of splitting record areas has serious drawbacks. If you are adding many records with similar keys (mass inserts), VSAM is forced to split many times. CI and especially CA splits are very time consuming. A file can become unusable if a split does not complete (ex: system crash). If you try to reduce CA splits by increasing CA%, you waste large amounts of disk space as most of the free CIs will never be used.

IAM CONCEPT

IAM's concept of Overflow is far superior to VSAM's. The IAM Overflow area is used for inserts, or record updates with an increase in length, which do not fit in the blocks where they should be added. If a block is full, IAM adds the record to Overflow. The IAM Overflow area is a record based area, which means that as records are deleted from the overflow area, the space can be reused for any other record, regardless of the key. An index to the Overflow area is kept in virtual storage, separate from the prime index. This concept makes much more efficient use of disk space because it is not tied to the file's existing index structure. For Enhanced format files, this Extended Overflow area is dynamically acquired on an as needed basis, acquiring and using additional DASD extents. Further, with the Variable Overflow feature available for Enhanced format files, IAM will make even more efficient use of the DASD space that is assigned to the Extended Overflow area.

For Compatible format files, the Independent Overflow area is allocated and formatted during the file load. The size of this area, which can be specified either by CA% free space, or by the OVERFLOW override, must be large enough to handle the volume of records anticipated to be inserted between file reorganizations.

IAM never moves data records once they have been placed on the file, unless the record increases in size and no longer fits in the block. In this case, IAM puts the record into Overflow first, before deleting it from the prime area. If an abend occurs, IAM retrieves the correct copy of the record bypassing the duplicate record on subsequent processing.

IAM files do not become unusable because of system crashes or job cancellation. Inserts into IAM files are much faster than VSAM.

It is very easy to tell how many Independent Overflow records have been used or remaining. The IAMINFO report gives you the exact number of records currently in Independent Overflow and how many records are left for additions. It is next to impossible to get this information from VSAM.

03.10 ENHANCED FORMAT FILE STRUCTURE

FILE STRUCTURE OVERVIEW

The IAM Enhanced File Structure has been designed to provide a base for the future enhancements and capabilities. The advantages of the Enhanced File Structure over the Compatible File structure include:

- Ability to take secondary extents after the file is loaded. The size and management of the overflow and prime extension areas are dynamically managed by IAM.
- Variable Overflow that manages space within overflow blocks as containing true variable length records. This is unlike the Compatible file format which only stores in each overflow block the number of maximum length records that will fit.
- Significantly larger number of buffers per file can be allocated, offering greater I/O savings.
- Alternate Index Support
- Single System Record Level Sharing Support
- Support of IBM hardware compression instruction
- Support of large devices (greater than 32K tracks per volume), which includes allowing up to 123 extents per volume
- Support for Parallel Access Volumes

All records contained within the IAM Enhanced File Structure are treated internally as variable length records. Additionally, all files that are defined as 75 tracks or larger will be automatically eligible for the IAM data compression, which helps reduce DASD space and provides I/O performance benefits.

IAM files are self-defined, non-VSAM files. This means that the information about the file structure is contained within the file itself, rather than in the system catalog and VVDS, as VSAM files are. The information kept within these system control areas, the system catalog, the VTOC, and the VVDS for DFSMS managed volumes is the same as for any other non-VSAM file. IAM files are treated as non-VSAM files by most DASD management products, including the FDR family of products.

An IAM file, as it appears on DASD, consists of a set of unique areas. Each area will be explained in further detail below. The basic structure is:

1. File Definition and Structure Data (Two blocks)
2. Prime Data Area
3. Index to Prime Data Area
4. Customize Hardware Compression Dictionary (if applicable)
5. Extended File Area
   - Extended area index and contents description
   - Extended Overflow and/or Extended PE blocks

03.10 CONTINUED . . .

FILE DEFINITION AND STRUCTURE DATA

The first block is initialized when an IAM file is defined with the basic file definition information, such as record length, key size, key offset, and so forth. With the IAM Alternate Index support, the information on related datasets (alternate indexes and paths) is also kept in the first block. After a file has been successfully loaded, the last I/O done to the file is to store the file structure data in the first block. Included is an indication that the file has been successfully loaded, and of course information about where the index begins, size of index, plus additional required data. The only updates done to this block after a file load are for statistical information that is written during file close processing, including information on number of inserts, deletes, and updates. The bulk of the information that is presented on the IAM Listcat report is taken from this control block. The statistical information, and information about the extended file area that are kept in the first block may not be accurate, and do not have to be accurate as they are for informational purposes only. A listcat that is done while the file is open to any application program will not reflect the exact status of the file, in much the same way as a listcat on an open VSAM file will also not reflect the exact status and statistics for the file. The statistical information can become inaccurate if a system or address space fails while the file is opened for update processing, and has not been successfully closed.

The second block contains information about the extended file area, including maximum size, and the location, by relative block, of where the information about the extended area is stored. This block is typically read during open processing, and is updated immediately after a new DASD extent has been acquired, and during close processing. If the file has not been successfully closed, then open processing will detect any inconsistencies, and update this data with the correct information.

PRIME DATA AREA

The Prime Data Area is built as the file is being loaded. This area contains the data records that were passed to IAM during the file load process. There is included some imbedded freespace within each block, called the Integrated Overflow area. This area is similar in concept to the VSAM CI freespace, and the size is indicated by the CI% Freespace parameter on the IDCAMS file DEFINE. Every loaded IAM file has a Prime Data Area, except for files that have been loaded with a single record.

INDEX TO PRIME DATA AREA

This index is built during close processing for a file load. This index consists of the high key in each prime data block, and may be in a compressed format. Once a file is loaded, this index structure is never changed, until the file is reorganized or reloaded.

HARDWARE COMPRESSION DICTIONARY

For datasets that are loaded with a user specified hardware compression dictionary, a copy of the dictionary is stored in the IAM file after the prime index. This is done to insure that IAM will be able to decompress the data in the file, even if the user subsequently changes the dictionary.

EXTENDED FILE AREA

The Extended File Area consists of the data that is added to the file, either as inserted records or from updates that increased the size of data records, which could not fit into the Prime Data Area of the file. This area consists of blocks containing control information about the Extended file area, and extended data blocks, which can be either Extended Overflow, or Prime Extension (PE). The control information for the Extended File area is based on the internal logical structure of the file, and is not necessarily tied to actual DASD extents. This way, IAM files can have their extents merged and eliminated by products such as FDR/CPK without impacting the integrity of the IAM file structure or the data it contains.

03.10 CONTINUED . . .

DYNAMIC FILE EXPANSION

The Extended File Area is acquired and formatted as needed. When there is a need to acquire an additional data block, for either Overflow or PE, a segment of the allocated and unused space is formatted. Normally, for batch processing, up to one cylinder will be formatted with empty blocks. For online processing, normally only one track will be preformatted at a time, to reduce the impact on response time. Blocks are then assigned as needed to either Extended Overflow, or PE. Once all of the allocated space is used, additional DASD space will be requested through the normal MVS EOV service.

If an error condition occurs during the EOV processing, such as an X37 abend condition, it is captured by the IAM DCB ABEND exit to prevent the job from actually abending. The request requiring the additional DASD space is failed, with a file full logical error. The avoidance of the abend is done to be compatible with VSAM, which will not abend either. The user will see the IBM error messages relating to the error condition encountered and an IAMW13 File full error message.

When a new extent is acquired, additional Extended Area control blocks are formatted and written as necessary, and up to one track of empty blocks will also be formatted. The control information for the Extended File area consists of identifying the blocks that are assigned to Extended Overflow, and the blocks that are assigned to PE. Additionally, the high key for each assigned PE block is retained as the index for the PE area.

EXTENDED PE BLOCKS

As records are being added to the logical end of the file, defined as having keys higher than what have been previously loaded or added to the dataset, PE blocks are assigned from the Extended File Area. Once a PE block is considered full, then the high key in that block is used as the index entry. Just as with the Prime Data blocks, Integrated Freespace, or CI% Freespace, is left in each block. This will allow for records increasing in size, as well as for later record insertions. Once a block is assigned as PE, it will remain as a PE block until the file is reorganized or reloaded. A PE block is only able to hold records that fall into the established index key range. If records are deleted from the PE block, the freespace is available for expansion of existing records within that block, or for new records added to the file within the established key range.

EXTENDED OVERFLOW BLOCKS

The IAM Extended Overflow area is a record based overflow area, similar to the Independent Overflow area of the Compatible File Format. Extended Overflow blocks are used to handle records that are being inserted within the file when there is insufficient space within the Prime Data or Prime Extension block that the record would have been assigned to, based on the established index. Extended Overflow space will also be used when an updated data record increases in size, and there is insufficient room within the block that it currently resides in for the larger record. The format of the data in the Extended Overflow block is identical to the data in the prime blocks. With Variable Overflow enabled, IAM will fully utilize the space available within each block. Without Variable Overflow, the number of records actually stored in the block will be limited to the number of maximum size records that will fit. When a record is deleted from Extended Overflow, the space it occupied is immediately available for reuse by ANY inserted or updated record, regardless of the key value. This eliminates the unusable lost space condition that can occur within VSAM files taking CI/CA splits, as records are deleted from certain key ranges, and new records are added in different key ranges.

The index to the Extended Overflow area is record based, that is each record in Extended Overflow has an entry in the index, consisting of the key and the block number of the Extended Overflow block containing the record. There will be an index to overflow for each of the prime data or extended PE blocks that have associated records in overflow. This enables IAM to use a compressed key structure for the overflow area, as well as potential reductions in CPU time to build and manage the overflow index when there are a very large number of records in the overflow area. The index is built when the file is opened, by reading all of the used Extended Overflow blocks, as indicated by the control information.

03.20 COMPATIBLE FORMAT FILE STRUCTURE

This section describes the Compatible format IAM file. An IAM file does not exist until a load process has initialized the space allocated for the file. Within the space allocated for a file, IAM establishes five distinct areas. These areas are the foundation for management of the file. The following is a description of how these areas are structured.

CONTROL AREA

The first area established is IAM's file definition and control area. Control information within an IAM file includes such things as:

1. The file characteristics (blocksize, number of blocks in the file, etc.)
2. Logical record description (record length, relative key position, etc.)
3. Overflow specifications (type of overflow, number of records, etc.)
4. Execution time options (core usage, key compression, I/O buffering etc.)
5. A control area ID, which serves as an indication that the file has been successfully loaded.

INDEPENDENT OVERFLOW AREA

The next area established is Independent Overflow. During file load, IAM establishes an area in the file capable of accepting the Independent Overflow records this file was defined to accommodate. The size of this area is calculated based on the number of Independent Overflow records requested and the block size of the IAM file.

When an IAM file is defined, the FREESPACE CA% value is used to calculate the size of Independent Overflow. IAM calculates its Independent Overflow value using the CA% and primary allocation values. Since IAM makes more efficient use of overflow, the CA% specified is cut in half for files with more than 10 cylinders.

Based on the size of the file's primary allocation, IAM will not reduce the CA% below the values shown in the following table:

Primary Allocation	Minimum CA%
1 cylinder or less	**10%**
2 to 3 cylinders	**9%**
4 to 5 cylinders	**8%**
6 to 7 cylinders	**7%**
8 to 9 cylinders	**6%**
10 cylinders or more	**5%**

Figure 1: Independent Overflow and CA% Freespace

IAM calculates the approximate number of maximum size records that will fit in the primary allocation and multiplies the result by CA percent. This value will become the number of records in Independent Overflow.

The maximum number of records that IAM will reserve based on CA% freespace is from the IAM Global Options Table MAXOVERFLOW value, which defaults to 50,000. The minimum is the lower of 500 records, or the number of records that will fit in half of the primary allocation. The number of Overflow records IAM is to reserve in a file can be overridden, when it is defined and at execution time when it is loaded, using an IAM CREATE Override Statement.

03.20 CONTINUED . . .

INDEPENDENT OVERFLOW AREA (CONTINUED)

IAM recognizes SMP/E CSI files by their cluster name (CSI in the last index level) and will use a default of 20% Integrated Overflow and 50,000 Independent Overflow records.

It is very easy to tell how many Independent Overflow records are used or remaining in an IAM file. The IAMINFO report or a LISTCAT ALL with IAMPRINT gives you the exact number of records currently in Independent Overflow and how many records are left for additions. It is next to impossible to get this information from VSAM.

If an application loads just one (1) record to a file, IAM formats additional space for Independent Overflow within the primary space allocation. The total combined space for Prime Extension (described below) and Independent overflow is approximately 90% of the primary space requested.

PRIME DATA AREA

The next area is called the Prime Data area. When the file is loaded (or created), IAM accepts records to build the file's prime data area. Records passed to IAM load must be in ascending key sequence and are placed into the prime data blocks. During the file load IAM reserves, as free space within every prime data block, the percentage of Integrated Overflow (CI%) specified for the file. IAM continues the prime area load until the user application or IDCAMS REPRO stops passing records and Closes the file.

PRIME EXTENSION AREA

The Prime Extension area is established following the prime data blocks. It is used to accept inserts to the file with keys higher than the previously existing high key on the file. Prime Extension is specified in blocks. Use the keyword PE= in the IAM Override statement to change the Prime Extension value.

If an application loads just one (1) record to a file, IAM formats additional space for Prime Extension within the primary space allocation. The total combined space for Prime Extension and Independent Overflow will be approximately 90% of the primary space requested.

INDEX AREA

The next area established is for the Index blocks. IAM establishes a high key value for each prime data block during file load. IAM uses one key per block for the prime index. These key values are written out to a data space under OS/390 or MVS/ESA, or as a temporary work file under MVS/XA, as each prime block is loaded. During CLOSE processing, these key values are read from the data space and used to create an index for the IAM file. When a data space is not used, IAM dynamically allocates the temporary work file.

IAM compresses the keys in the index using a proprietary compression technique on each key, which results in a higher compression ratio than VSAM. If the size of the compressed index is at least 10% smaller than the uncompressed index, the compressed key structure is stored in the file after the uncompressed index.

10.01 USERS GUIDE INTRODUCTION

OVERVIEW

Welcome to the IAM Version 8.0 Users Guide. This guide is designed to explain how to use IAM Enhanced Format datasets. It is primarily a task oriented guide. For each task, general guidance and reference information is provided, along with numerous examples to aid in the understanding and use of IAM.

This guide assumes that the reader has a general understanding of VSAM and the related terminology, has a working knowledge of OS/390 JCL, and the Access Method Services (IDCAMS) utility program. Additionally, the reader should be familiar with the IAM Features and Capabilities as described previously in this manual. The primary focus of this Users Guide is on how to, as opposed to explaining each IAM or VSAM feature and concept.

VSAM COMPATIBILITY

IAM is a high performance indexed access method, providing random and sequential access to user data with minimal computer resources. IAM provides an application program interface that is fully compatible to the OS/390 VSAM access method, supporting the most commonly used features and capabilities. IAM can be used in place of VSAM KSDS files, which are processed sequentially or randomly by key, and VSAM ESDS files, that are accessed sequentially or randomly by relative byte address, or by control interval. IAM files can be used in place of VSAM KSDS or ESDS files that utilize the IAM supported functions without modification. IAM, with the optional alternate index feature, can be used for any alternate indexes related to VSAM KSDS or ESDS types of file that have been converted to IAM. While IAM does not use the VSAM LSR buffer pool, IAM can be used by applications that indicate usage of the VSAM LSR buffering, including Batch LSR and CICS. IAM files can be processed by system utility programs, including IDCAMS and any of the many SORT software products. IAM provides support for the VSAM exit routines, as specified in the ACB EXLST, including the SYNAD, LERAD, EODAD, JRNAD, and UPAD exit types.

IAM does not support accessing KSDS type of files by VSAM relative byte addressing or by control interval processing (VSAM RPL OPTCD=ADR or OPTCD=CNV). IAM does not support control interval updates for ESDS type of files, although control interval reads and file load processing by control interval are supported.

USING IAM DATASETS

IAM files are processed in a manner that is identical to VSAM. First, IAM files must be defined. IAM files can be defined with the IDCAMS DEFINE utility, through MVS JCL VSAM allocation, or by the IAM ISPF panels. Once defined, IAM files must be initialized with user data. IAM files can be loaded with application programs, IDCAMS REPRO, or as system SORT output (SORTOUT). After a successful file load, IAM files can then be processed and updated. Alternate indexes and paths can be defined and built using IDCAMS. After significant update activity, an IAM file may need to be reorganized, just as VSAM KSDS files. The frequency of file reorganization of IAM files may be either less or more than what was required with VSAM. Innovation offers a file reorganization product, called FDRREORG™, which can automate the file reorganization process for IAM files, as well as VSAM and PDS type of files.

The subsequent sections will present information and examples on how to perform all of those tasks, as well as other dataset management tasks. Before getting into the specific tasks, some general JCL guidelines are presented. While there are usually no JCL changes required to utilize IAM files, there are some additional IAM unique JCL statements that, when used, can enhance the overall IAM file usage. Such JCL changes provide for overriding various IAM default options, can provide IAM unique reports, and IAM tracing and debugging capabilities.

10.10 JCL CONSIDERATIONS

As a general rule, there are no JCL changes needed to use IAM files. JCL parameters for VSAM, such as the VSAM AMP parameters can remain without any need for change. IAM will ignore any value for BUFNI. IAM will honor the STRNO parameter, which specifies the number of place holders. IAM will use the value specified by the BUFND parameter for the maximum number of buffers, MAXBUFNO, unless it is less than the default maximum buffers, or a MAXBUFNO override has been specified. Files with JCL for Batch LSR can remain so specified, although IAM will not use the LSR buffer pool.

IAM UNIQUE DD CARDS There are some unique IAM DD statements, which are described below. The use of the below mentioned DD names is generally optional, however they may be requested by IAM Technical Support for gathering additional data for problem diagnosis.

DDNAME	Description
IAMDEBUG	When specified as a DD DUMMY, will result in IAM issuing a U0184 ABEND for various error conditions. This is done to provide a simple mechanism to obtain a dump for problem diagnosis. This DD statement will also cause various IAM error messages to appear, which normally would not appear because of commonly encountered error conditions that are typically handled by return codes. **The IAMDEBUG DD statement should only be provided when requested by IAM Technical Support because a normally running job may abend unexpectedly.**
IAMDUMP	An optional DD card, that when specified, will be used for dumping the IAM index data space when an IAM file is being closed under a task that is abending. This will normally be a SYSOUT dataset.
IAMINFO	Identifies a sequential output dataset, which is normally SYSOUT. IAM will set the required DCB characteristics for this dataset, so specification of a DCB should NOT be done. Presently, IAM produces records with a RECFM=FBA and an LRECL=121 for the IAMINFO report. Caution should be used when specifying large block sizes for this dataset. A typical IAMINFO report contains about 45 lines or records, so block sizes larger than 5445 are not useful. This dataset will contain a report that is produced by IAM every time an IAM file is closed. The report contains file description information, along with statistics on file usage and resource requirements to process the dataset. Use of this feature is highly recommended, and may be requested by IAM Technical Support for problem resolution, particularly for performance related questions. These IAMINFO reports can also be generated by the IAMSMF utility program, provided the installation is collecting and saving the optional IAM SMF records.
IAMNOLIC	When specified as a DD DUMMY on job steps that perform an IDCAMS LISTCAT requests, will prevent the production of the IAMPRINT reports for IAM files. This is primarily intended for use when a job is doing a LISTCAT of an entire catalog, and it is desirable to eliminate the overhead of IAM processing, which can be significant when thousands of datasets are being processed.
IAMOVRID	Specifies a card image dataset, which contains the control cards and keywords to use or modify the use of various IAM features. This is normally a DD * (SYSIN type) of dataset, however it can also be a sequential dataset, or a member of a partitioned dataset. Refer to the section on IAM Overrides for complete information on using this facility.

10.10 CONTINUED . . .

IAMPRINT Identifies a sequential output dataset, normally SYSOUT, that contains a report on IAM file characteristics whenever a LISTCAT ALL is done for an IAM file. IAM will normally dynamically allocate this file to the default SYSOUT class (SYSOUT=*). For TSO users, the output is automatically routed to the user's terminal. The DD card can be optionally provided by the user, to capture the output to a dataset, or route the output to a different SYSOUT class. The user need not provide any DCB information, however should be aware that the file will have an LRECL=121 and a RECFM=FBA.

IAMWKDD An optional DD statement that provides a temporary dataset to be used during an IAM file load to hold the index. Normally, IAM will default to using a Data Space of up to 256 megabytes that is adequate for all except the extremely large dataset. For example datasets that are using _ track for a block size, with a 64 byte key, the data space is sufficient for up to 51 gigabytes of compressed data. To force the use of the work file, the CREATE Override DATASPACE=0 must be specified.

$NOVIF When specified within a job step, indicates that IAM will not flag dynamically allocated IAM datasets as DSORG=VS in the JFCB for the dataset. This is primarily for use by dataset management utility software, such as CA-DISK.

10.20 DEFINING IAM FILES

OVERVIEW Before using an IAM dataset, it must be defined. This define process is identical to what is required for VSAM datasets. During the define, IAM allocates the DASD space for the dataset, catalogs the dataset, and stores the file attributes within the dataset itself. IAM datasets can be defined by using IDCAMS, through JCL DD cards, or using a variety of methods under TSO, including through the IAM ISPF panels. Many other software products that are used to define VSAM datasets will generally also be able to define IAM datasets. This section will provide information on parameters and examples to define IAM datasets using with IDCAMS, through JCL, and under TSO.

HOW TO IAM A DATASET For a dataset to become IAM instead of VSAM, an indication must be provided on the file definition indicating that the file is to be an IAM file. The ways to indicate this are:

1. Add the parameter OWNER($IAM) to the IDCAMS DEFINE command, or
2. Change the dataset name to include the literal $IAM somewhere within the name, or
3. For SMS managed IAM datasets, use an SMS Data Class or Storage Class with the literal $IAM as part of the class name.

For most datasets, all that is required to implement IAM is to change the file definition using one of the above techniques. Any of the above methods can be specified on an IDCAMS DEFINE. For JCL definition of an IAM dataset, the OWNER parameter is not available, so either the dataset name has to include $IAM, or it must be a part of the dataset's Data Class or Storage Class name. Most installations select one method as their preferred method, based upon their internal dataset management or accounting requirements. While many installations have chosen the OWNER($IAM) technique, the alternative of placing $IAM within the dataset name has the advantage of making identification of IAM datasets very easy and many installations have chosen this route as well.

BASIC DEFINE PARAMETERS For any IAM dataset, there is certain basic information that must be provided. This is usually provided through keywords specified either on the IDCAMS DEFINE command, or as JCL. Other sources of these attributes are an SMS Data Class, or from another IAM or VSAM dataset as a model. The basic required information for all types of IAM datasets include:

- Dataset Name
- Indication that file is to be an IAM file, e.g. OWNER($IAM)
- Volume(s) on which dataset is to reside
- Quantity of DASD space required
- Maximum record size
- Type of dataset (i.e., KSDS or ESDS)
- Key length and offset (RKP) for KSDS (INDEXED) type of files

Additional information that can be provided includes free space, share options, and expiration date. Several of the other VSAM file attributes can be specified, such as IMBED, REPLICATE, SPEED, etc., however they are not relevant to an IAM dataset and will be ignored. Certain attributes unique to IAM can be specified via IAM overrides, which include data compression, IAM file format, and default buffering range. The various unique IAM attributes can also be set as installation defaults in the IAM Global Options Table.

If the IAM allocation encounters any errors, the error messages will appear on the JES job log, with the MVS allocation messages (SYSMSGS) and on the IDCAMS SYSPRINT, if it is available. Due to the manner in which IDCAMS prints messages on SYSPRINT, the error messages from IAM will precede the DEFINE command. IDCAMS will also print out additional error messages after the DEFINE, performing an analysis on the return codes set by IAM. Whenever possible, IAM uses the VSAM return codes that most clearly indicate the actual problem, although that is not always possible. Always refer to the IAM and related allocation error messages for the most precise problem determination possible.

10.20 CONTINUED . . .

CONSIDERATIONS FOR DEFINING AN IAM DATASET

In general one can easily convert a VSAM cluster to IAM just by modifying the DEFINE, as described above. Because IAM has a different file structure, and is allocated as a non-VSAM dataset, there are some differences between IAM dataset allocations and VSAM allocations that may affect a few of your datasets.

DASD SPACE CONSIDERATION

IAM datasets typically require from 30 to 70% less disk space than your existing VSAM clusters. IAM datasets use DASD space more efficiently. The compressed index and advanced internal structure usually result in about a 20 to 40% reduction of disk space compared to a similar VSAM cluster. IAM's data compression may provide an additional 20 to 50% reduction in disk space for most datasets. Customers with IBM z/Series processors can also consider using the IAM hardware compression instead of the software compression. Using the hardware compression, along with an optional customized compression dictionary, may yield even greater space savings.

In an effort to conserve disk space and prevent over allocation, IAM releases space that is unused and not being reserved after the file is initially loaded. This is done automatically when secondary space is specified. If you want to override IAM's default of releasing the over allocated space, see IAM Override statements. The keyword is RELEASE=NO. After a file has been loaded, a LISTCAT ALL will show you the exact number of tracks an IAM file is using and has allocated. Innovation recommends that when converting VSAM files to IAM, initially retain the original VSAM space allocation values. After observing the IAM space requirements, the space allocation can be adjusted if so desired.

IAM datasets exist on DASD as either a non-VSAM type of dataset or a DFSMS extended format sequential dataset. IAM files that are not DFSMS extended format abide by non-VSAM rules for secondary allocations. One major difference is that these IAM files can only have a maximum of sixteen extents per volume. To compensate for this limitation, IAM has a capability to automatically increase the secondary space value. This can be controlled through the MAXSECONDARY IAM Override and Global Option, which is a multiplication factor on the specified secondary space quantity. Using this, IAM will increase the secondary space allocation up to the limit set by MAXSECONDARY, so that IAM should be able to obtain approximately the same amount of DASD space as VSAM. These non-extended format IAM files are also limited to 64k tracks per volume.

IAM datasets that are DFSMS extended format datasets can have up to 123 extents per volume and can use a maximum of 64k cylinders per volume. To obtain an extended format dataset, the DFSMS data class (DATACLASS) must specify "Dataset Name Type" of "EXTENDED REQUIRED or EXTENDED PREFERRED, and have a storage class (STORCLASS) with a Sustained Data Rate (SDR) of either blank or 0. The disadvantage of using extended format datasets is that each block has an additional 32 bytes of system data appended to it, so such datasets may use more DASD space than when in not in a DFSMS extended format.

10.20 CONTINUED . . .

MULTIVOLUME CONSIDERATION

IAM utilizes standard MVS services to acquire additional DASD space. Because IAM datasets are non-VSAM, the rules and mechanisms for acquiring additional space for multi-volume datasets are different than VSAM. When IAM needs additional space, it issues the MVS EOV (End of Volume) service to acquire additional DASD space. The only input IAM can provide is a space quantity, by specifying the desired secondary quantity. IAM will attempt adjustments on the secondary quantity, as per the MAXSECONDARY and MULTIVOLUME parameters.

The basic rules for IAM datasets are that the primary space quantity has to be available on the first volume. Then, as new extents are acquired, generally additional space will be acquired on the current volume, until the dataset has sixteen extents on that volume, or there is insufficient space to satisfy the request. Then, MVS will switch to the next candidate volume. When the next candidate volume is explicitly named, which is typical for datasets not managed by SMS, there must be sufficient space on the next candidate volume for the requested secondary quantity. In other words, MVS non-VSAM EOV can not skip over a "candidate" volume due to insufficient space and go on to the next. Attempting to do so will cause a SE37-08, resulting in an IAMW13 File full message.

There are some exceptions to the above described processing. If the file is SMS managed in a guaranteed space Storage Class, then the primary allocation is made on each volume when the dataset is defined. If the secondary space is zero, then once the primary allocation is used on one volume, IAM will be switched to the next volume. When all the allocated space is used, attempts to add more data will fail due to a file full error. When a secondary quantity is specified, then additional extents will be acquired on the current volume, providing there is space to do so, until the dataset has reached either the limit of 16 extents on that volume, or has run out of space. Then, IAM will be switched to the next volume. IAM datasets can not exceed a total of 255 extents.

A second exception occurs when the IAM file is a DFSMS Extended Format dataset. The DFSMS Extended Format datasets can have up to 123 extents per volume, and can exceed the total of 255 extents. Therefore IAM datasets that are DFSMS Extended Format will not be subject to the IAM space adjustment function. The DFSMS Extended Format datasets are still limited to a maximum of 59 volumes.

A third exception occurs when the file is defined as multi-volume, not SMS managed and the user has specified a secondary quantity of zero. For systems that have SMS active, IAM will treat this type of allocation like an SMS guaranteed space request. The primary quantity is allocated on each volume when the file is defined. When all the allocated space is used on one volume, then the allocated space on the next volume will be used. Unfortunately, this technique does not work on systems that do not have SMS active. So, for those systems, IAM will set the secondary to be the same as the primary. This usually results in only the primary space being allocated on the first volume, and then potentially multiple extents on the second and subsequent volumes.

Reorganizing, or reloading a multi-volume IAM dataset without deleting and redefining is not recommended because that can result in some strange space distribution across the volumes. This is because of the processing done by MVS EOV when using a previously existing file. What happens is MVS will cause all of the currently allocated extents to be used first. MVS will not acquire new extents on any volume which has extents already, unless it is on the last volume. New extents, if needed, start with the first unused candidate volume. This can result in some volumes not having as much space utilized as desired or needed, causing unnecessary file full conditions.

DFSMS SUPPORT

IAM provides full support for SMS managed IAM datasets. By definition, to be eligible for an SMS managed volume, the dataset must be assigned a Storage Class. The DATACLASS, MGMTCLASS, and STORCLASS can be explicitly provided on the DEFINE command, or selected by the ACS routines. The Data Class can provide file characteristics for the file being defined, including record length, key length, key offset, share options, free space, and others, eliminating the need to specify those values explicitly on the DEFINE. As per SMS rules, the options in the Data Class will be used, unless explicitly overridden on the DEFINE command. IAM files on SMS managed volumes will be cataloged with the class names.

10.20 CONTINUED . . .

DFSMS ACS ROUTINES

One special consideration for IAM files in an SMS environment is that the ACS routines will be invoked twice for the same request. The exceptions to this are for files being allocated through JCL, and files being allocated through the IAM ISPF panels. This is because IAM DEFINE intercept occurs after IDCAMS has invoked SMS to analyze the DEFINE request. Subsequently, if the file is going to become an IAM file, IAM will issue a dynamic allocation for the dataset, which will cause the ACS routines to be invoked again. When IAM issues the dynamic allocation request, IAM will specify whatever SMS classes have been assigned to the dataset at the point in time that the intercept occurred.

One of the problems that can occur is if an installation has established an SMS Storage class, such as STORCLASS(NONSMS), which users can code to prevent a dataset from being SMS managed. That storage class name is nullified on the first pass through the ACS routines, causing the file to be unmanaged and preventing IAM from seeing that original storage class specification. When the ACS routines are called again out of the dynamic allocation issue by IAM, because the STORCLAS is null, the ACS routines assign the dataset to an SMS managed STORCLAS. This problem is resolved by the use of the IAM STORCLAS Global Option. By setting that option to be the non-managed storage class, (e.g. NONSMS), IAM will pass that as the STORCLAS on any allocations which do not have one at the time IAM intercepts the request.

Another problem that can occur is if the ACS routines decide on the second pass that the dataset is not to be a DFSMS managed dataset. This will cause the dynamic allocation to fail, because the volumes that are passed are DFSMS managed volumes. To prevent this from occurring, the ACS routine must not nullify that STORCLAS for an IAM file that has a STORCLAS specified. The STORCLAS can be changed to a different STORCLAS if desired.

A third consideration for the ACS routines is that some installations have set their ACS routines to perform different actions if some of the classes are already specified on entry to the routine. This is usually due to installations wanting to limit the external use of SMS classes by their users. Because of this, ACS routines with code that checks for the preexistence of SMS classes, and performing different actions, could result in IAM files being assigned to a different classes than expected or desired.

The main point here is that the developer of the ACS routines, particularly the Storage Class routine, must be aware of how IAM file allocations work, and to code the routines to achieve their installation's desired results with the above considerations in mind. Establishing an installation standard to use $IAM in the dataset name, or as part of a user specified DATACLAS will make it much easier to identify IAM files in the ACS routines. (Note that the OWNER parameter from the DEFINE is NOT accessible to the ACS routines.) The IAM technical support team is available to help review ACS routines, and make suggestions on revising them to meet their objectives for IAM files.

10.20 CONTINUED . . .

DFSMS EXTENDED FORMAT DATASETS

Starting with IAM Version 8.0, IAM datasets can be allocated as DFSMS Extended Format datasets. This attribute will allow IAM datasets to have up to 123 extents per volume, use up to 64k cylinders per volume, instead of only 64k tracks, and can exceed 255 total extents when defined on multiple volumes. These IAM datasets cannot be striped, that is they cannot have a stripe count greater than 1, and cannot be compressed by the DFSMS compression function. The IAM software or hardware compression can be used.

To define an IAM dataset as a DFSMS Extended Format, users must either explicitly assign a data class and storage class, or have the ACS routines provide a data class and storage class, with the following attributes:

1. The DATACLASS must have a “Dataset Name Type” of EXTENDED REQUIRED or EXTENDED PREFERRED.
2. The STORCLASS must have a “Sustained Data Rate” value of blanks (nulls) or 0. Setting other values will cause the dataset to be STRIPED, which IAM does not support.

A couple of notes on using IAM Extended Format datasets. First, DFSMS will always append 32 bytes of system data to each block. IAM automatically adjusts its block size calculations to consider this requirement, however the resulting dataset may require more DASD space than when it is kept in a non-extended format. Secondly, IAM uses BSAM WRITE macros when an Extended Format dataset is being loaded. Therefore, the EXCP counts for file loads of IAM Extended Format datasets will be the number of blocks written to DASD, not the actual number of EXCP’s issued, which will result in higher EXCP counts for file loads of these datasets. Innovation recommends that you only use DFSMS extended format datasets when a dataset requires one or more of the benefits provided by such a dataset, as describe previously.

NON-SPECIFIC VOLUME ALLOCATION

For datasets that are not managed by DFSMS, IAM offers allocation to non-specific volumes. If you wish to use the IAM non-specific allocation, specify VOLUME(ANYVOL). For multi-volume non-specific allocation, specify VOLUME(ANYVOL ANYV01 ANYV02 ...). This will result in IAM issuing a non-specific dynamic allocation request for the IAM file. The first volume will be selected by MVS allocation. Any additional volumes are selected by IAM, which will select volumes from the specified UNIT name that are of the same device type as the first volume selected. IAM builds a list of the eligible volumes, then selects those volumes that have the largest quantity of available contiguous space. All of the volumes must be mounted as STORAGE to be eligible for selection.

Volume selection is done at the time the dataset is defined, and a subsequent LISTCAT will show the volumes selected. The default UNIT name used is SYSDA. To change to a different UNIT, use the IAM CREATE override keyword UNIT=, or change the IAM Global Option WORKUNIT.

When using the IAM non-specific allocation, do NOT specify UNIQUE. Doing so will cause IDCAMS to attempt to allocate the nonexistent volumes. Similarly, if the MODEL parameter is specified, also specify the SUBALLOCATION parameter, because the MODEL parameter causes the UNIQUE attribute to be assigned.

Customers that are using the CA-ALLOCATE (formerly Sterling Software’s product SAMS also known as VAM) can not use the non-specific volume allocation feature of IAM. They instead should use the pooling and volume selection provided by that product.

10.20 CONTINUED . . .

IAM OVERRIDES FOR IAM DATASET DEFINITION

DEFINE OVERRIDES

There are various unique attributes for IAM datasets that can optionally be set when the dataset is defined through the use of the IAM Override facility. The IAM override facility is provided because there is no mechanism within the DEFINE CLUSTER command to pass this information. Included amongst the attributes that can be specified are IAM Data Compression, the IAM file format, and enabling variable overflow. Most of these attributes are based on defaults from the IAM Global Options Table, which are normally set up during the IAM product installation. With appropriate choices made during installation, there should be a very infrequent need to use the IAM Overrides during the file define. An overview of the IAM Override facility is presented here, while complete information is provided in Section 30 of the manual.

IAMOVRID DD STATEMENT

The use of the IAM Override facility is triggered by providing an IAMOVRID DD card in the JCL, which usually consists of in-stream card image input (i.e. //IAMOVRID DD *). This DD can also reference a card image sequential file on DASD, or a member of a PDS. Each control card contains a word indicating the type of override (i.e., CREATE or ACCESS), then after a blank is followed by one or more operands. For a file definition, use the CREATE override statement. The CREATE override requires either the DD= or the DSN= operand, which identifies the dataset to which the override is applicable. A value of DD=&ALLDD will apply the override to any IAM dataset that is not explicitly overridden.

The IAM Override facility can be used for the Define Cluster commands issued under IDCAMS or TSO. It does not work for defines done through JCL DD cards or the TSO ALLOC command. Various IAM overrides can be specified directly on the IAM ISPF panel. The IAM Override process will most likely work with other software products that can Define VSAM files. Examples of using the IAM Override facility for file definition are included in with the IDCAMS Define Cluster examples later in this chapter.

CREATE OVERRIDE OPERANDS

The following list contains the CREATE Override keywords which are applicable to the file definition process. For the most part, the CREATE Override keywords have the same meaning and implication when used during the file load, as they do during the file define. The underscored portion of the keywords indicates their minimum abbreviation. The keywords applicable to the define process are:

Keyword	Brief Description
BLKSIZE=nn	Specifies the block factor (1 - 15) or block size of the IAM dataset. The default value is based on the specified CI Size and maximum record size. Typically the default is the maximum block size that can be used to obtain 4 blocks per track.
COMPATIBLE	Specifies that IAM is to create a compatible format dataset. A compatible file format has overflow areas that are formatted when the file is loaded, and are fixed in size until the file is redefined or reorganized. The default is taken from the IAM Global Options Table, which as shipped is Enhanced format.
DATACOMPRESS= [YES|NO|HW]	Indicates whether IAM is to data compress this dataset. The default is based on the IAM Global Options Table, which as shipped is any dataset that is 75 tracks (5 cylinders) or larger is eligible for IAM software data compression. Specifying HW indicates that IAM is to use the IBM hardware compression instruction. IAM files do not have to be DFSMS managed to use the hardware compression feature with IAM.
DDNAME=	Specifies the dataset to which the override is applied. When used during a DEFINE process, the value specified MUST match the value specified for the FILE parameter. If this value is set to &ALLDD, then the overrides are applicable to any IAM file that is not otherwise explicitly overridden.

10.20 CONTINUED . . .

DICTIONARY= Specifies the four character suffix for the name of the user provided hardware compression dictionary. The dictionary must be in load module format, with the first four characters being 'IAMD'. Review the section on Hardware Compression for information on creating and naming the compression dictionary. The default is to use the IAM provided dictionary.

DSN= Specifies the name of the dataset to which the override is applied. Either DDNAME= or DSN= must be specified.

ENHANCED Specifies that IAM is to create an Enhanced format IAM dataset. This type of dataset uses a dynamic overflow area, which can acquire additional DASD extents during file updates, as needed. The default is taken from the IAM Global Options Table, which is shipped as Enhanced format.

INTEGRATED=nn Specifies CI% free space, especially useful for data compressed ESDS datasets.

MAXBUFNO=nnnn Specifies the default maximum number of buffers to be used during file access for this dataset.

MAXSECONDARY=nn Specifies a default value from 0 - 10 as a multiplication factor for the secondary space quantity, when the dataset exceeds five extents on a volume. When overridden on the DEFINE, this will be the default for both file load and file access for this dataset.

MINBUFNO=nnn Specifies the default minimum number of buffers to be used during file access for this dataset.

MULTIVOLUME= [PRIMARY | SECONDARY] Specifies space quantity to request when IAM believes that the next extent will be placed on the next candidate volume. The default is specified in the IAM Global Options Table, and as shipped is MULTIVOLUME=PRIMARY.

PSEUDORBA For ESDS files, indicates that the file can exceed 4 gigabytes of user data. IAM generates RBA values that are different than normal VSAM values. This attribute can not be used with SAP ESDS files, or other software that depends on the normal VSAM values.

RELEASE=[YES|NO] Indicates whether or not IAM is to release unused and unreserved DASD space whenever the file is loaded. Default is to release space on the first load only, and only when a secondary space value is specified.

UNIT= Specifies the unit name to specify when allocating the IAM dataset. The default is SYSDA for non-specific volume allocations, or the generic unit type, e.g. 3390.

VAROVERFLOW=[YES|NO] For Enhanced format files, indicates that IAM can use true variable length records in the Extended Overflow area. Variable Overflow provides a more efficient use of DASD space. The default is from the IAM Global Options table, which is shipped set to YES.

XESDS For ESDS files, indicate that IAM is to use an 8-byte RBA value. This is consistent with the VSAM Extended Addressability support. Default is that IAM will use a 4-byte RBA. The 8-byte RBA is a way for ESDS files to exceed 4 gigabytes of data, but application programs must be able to handle the 8-byte RBA.

10.20 CONTINUED . . .

DEFINING IAM DATASETS USING IDCAMS

IDCAMS DEFINE When the IAM VSAM Interface is active in the system, every DEFINE is analyzed by IAM. If the DEFINE contains the $IAM parameter, the file is created as an IAM dataset. If the $IAM parameter is not coded, a VSAM cluster is created.

IAM allocates a non-VSAM file on disk with a DSORG of PS. This dataset contains the index, the data records and the file characteristics all incorporated into a single dataset. IAM dataset must be cataloged in a VSAM or ICF Catalog.

The parameters specified in the DEFINE statement are applied to the IAM dataset. Because the IAM file and overflow structures are different from VSAM's, some of the parameters specified may be changed or ignored.

When the DEFINE is for an IAM dataset the following rules will apply:

1. The cluster name will be the name of the IAM dataset.
2. The data and index component names will be ignored.
3. Index component attributes are not needed and will be ignored.
4. Attribute parameters will be used or ignored as documented in the following paragraphs.

IDCAMS JCL CONSIDER-ATIONS There are normally no JCL changes to run IDCAMS to define IAM datasets, when converted from VSAM, unless IAM is not in the Link List or is using the IAM Override facility. If IAM is not in the Link List, then a JOBLIB or STEPLIB DD card will be necessary. Because IDCAMS operates as an APF authorized program, the IAM load library referenced by the JOBLIB or STEPLIB must be explicitly APF authorized in the IEAAPFxx member of SYS1.PARMLIB. If the library is not authorized, then MVS will not even look in the library for the IAM modules, which will result in a VSAM file being defined instead of an IAM file. The only other DD statements required are SYSIN DD for the control card input, and a SYSPRINT DD to SYSOUT. If, as in the examples provided, you do a LISTCAT ALL, then IAM will dynamically allocate an IAMPRINT DD to SYSOUT=*. If you do not want the IAMPRINT output to go to SYSOUT=*, then provide an explicit IAMPRINT DD card.

DEFINE WITH REPRO One word of caution must be given about specifying the IAM dataset on a DD card within the job step that the dataset is being defined. If the file is deleted and defined within the same execution, and is defined on different volume(s) than it was on originally, then the DD card will not reflect the newly defined allocation. The volume(s) for the DD card are allocated by MVS when the step is initiated, so if the dataset is moved to different volume(s) as a result of the define, attempting to use the DD card for I/O will result in errors. For example, if under a single execution of IDCAMS you have the following sequence of commands:

- REPRO INFILE(iamfile) OUTFILE(seqfile)
- DELETE my.iam.file
- DEFINE CLUSTER(NAME(my.iam.file) - etc.
- REPRO INFILE(seqfile) OUTFILE(iamfile)

you may experience errors when the second REPRO is done to reload the IAM file. This is particularly true if the IAM file is a DFSMS managed dataset, which will likely change the volume(s) on which the dataset resides. The circumvention for this problem is to change the second REPRO to use the OUTDATSET or ODS parameter instead of OUTFILE, and the first REPRO to use INDATASET or IDS instead of INFILE. This will result in IDCAMS dynamically allocating the dataset at the point in time the REPRO is being done, so the correct volume information will be available. Another alternative is to execute the REPRO in a subsequent job step. This problem is not unique to IAM, it can occur with VSAM as well.

10.20 CONTINUED . . .

BASIC PARAMETERS Most VSAM DEFINE parameters are applicable to IAM. The following paragraphs will document the parameters that are necessary for IAM, the ones that do not apply and any differences between IAM and VSAM.

The following parameters are for the essential information. Generally, these will be explicitly provided on the DEFINE, however the information other than the name can be filled in by using a MODEL of an IAM or VSAM dataset, or from an SMS Data Class.

Essential Keyword	Description
CLUSTER(	Identifies that a VSAM type dataset is to be created. This is a required parameter for an IAM dataset.
NAME(dsname)	This is a required parameter for an IAM dataset. The entry name specified for the cluster will be the name of the IAM dataset. The data and index component names are ignored. If '$IAM' appears anywhere in the name the dataset will be created as an IAM dataset.
OWNERID($IAM)	If OWNER**($IAM)** is coded on the CLUSTER statement, the file will be created as an IAM dataset. When '$IAM' is not present in the cluster name, and the $IAM parameter is not otherwise specified, the file will be created as a VSAM cluster.
CYLINDERS(xx yy) **TRACKS(xx yy)** **RECORDS(xx yy)** **MEGABYTES(xx yy)** **KILOBYTES(xx yy)**	Required information that indicates the amount of DASD space to be allocated for the IAM dataset. The unit of allocation is based on the keyword specified. The first value provided indicates the amount of space to be allocated during the file definition process. For IAM files, the primary quantity MUST be available on the first volume, otherwise the request will fail. The second value, which is optional, indicates the amount of additional DASD space to request in case the primary quantity is insufficient. The secondary quantity will be used to acquire additional extents during file loads or reorganizations. The secondary quantity will also be used to acquire additional space for Enhanced format IAM datasets as needed to handle record updates and inserts after the file load. If the IAM dataset being defined is not a DFSMS Extended Format dataset, then it is limited to 16 extents per volume, and a total maximum of 255 extents. The maximum extent size is 64K-1 tracks (65,535), or 4,369 cylinders per volume, which is also the maximum amount of space that can be used by IAM on any single volume. These limits do not apply to DFSMS Extended Format datasets.
VOLUMES(volser….)	Specifies the volume(s) on which IAM is to allocate the dataset. Due to the TIOT limitation, IAM files can reside on a maximum of 59 volumes. The volume(s) provided can be specific volumes, DFSMS non-specific volume of *, or non-DFSMS non-specific volume of ANYVOL.
KEYS(length offset)	For KSDS files, required parameter that specifies the length of the key, and the relative position of the key within the record. The maximum key length supported by IAM is 249. The maximum key offset is 4091. Because IAM does not data compress the bytes preceeding the key, it is beneficial to place the key as close to an offset of 0 as possible. The IDCAMS default values are: 64,0.

10.20 CONTINUED . . .

RECORDSIZE (average ,maximum) The first value specifies the average record length in bytes. The second value is the maximum record length. The maximum record size supported by IAM Enhanced Format files is almost 8 megabytes. There is not a specific maximum record size supported by IAM, as the maximum size possible is dependent on several factors, including the block size being used for the file, the key length, and the key offset. The DEFINE request will fail if IAM calculates that it can not support the specified maximum record size. Files with large record sizes that exceed the half-track block size on the type of device the file is being defined on must be defined with the SPANNED attribute.

The average record length is primarily used when the define request specifies that the space is to be allocated by RECORDS. Then IAM (and VSAM) will use the average record length when converting the space value to a device specific quantity.

Users of compatible format IAM datasets should review the section on considerations for compatible mode files for additional information on the implications on the file structure of this parameter. Because compatible mode files do not support spanned records, the maximum record size is 32,755 for variable length record files, and 32,760 for fixed length record files.

IDCAMS default values are: average=4089 maximum=4089

INDEXED NONINDEXED Specifies the type of dataset that is being defined. INDEXED indicates that a KSDS type of dataset is being defined, and NONINDEXED indicates an ESDS type of dataset.

IDCAMS default is INDEXED.

10.20 CONTINUED . . .

OPTIONAL PARAMETERS While the following keywords are all optional, in various situations they may be required or quite beneficial. Some of the keywords listed here can not be used for IAM files, and are presented for that reason. For ease of reference, they are presented in alphabetical sequence.

Optional Keyword	Description
BUFFERSPACE(bytes)	Specifies the maximum amount of virtual storage to be used for buffers for this dataset. IAM will use this value to calculate the effective MAXBUFNO for accessing the dataset, providing it does not go below the default value from the IAM Global Options Table.
CONTROLINTERVALSIZE (size)	For VSAM, this controls the logical and physical block size on DASD for the file. VSAM restricts the size to multiples of 512 or 2048 that usually results in underutilized disk space. IAM takes the size specified and based on the device type being allocated to and the record size, calculates a larger blocksize for the dataset that is a proportion of the track capacity. If the blocksize developed would result in more blocks per track than the default block factor, which is shipped as 4 blocks on a track, IAM will increase the blocksize until only 4 blocks will fit on a track. The blocksize may be further adjusted if the block cannot contain at least 4 maximum size records. The default block factor can be changed through the use of IAMZAPOP, by setting the VSAMBLOCKF parameter. The block factor or size can also be set by the B= operand of the IAM CREATE Override. For IAM ESDS files, IAM will use the specified CI SIZE in the calculations for the RBA of each record. When reorganizing an IAM ESDS, users should be careful not to change the CI SIZE, because it can result in different RBA values for the records. This may cause problems for applications that have an index into the ESDS file by RBA. IAM does store the specified CI SIZE, so that it is available information if the dataset is converted back to VSAM. EXAMPLE OF BLOCK SIZE ON A 3390: CISZ(4096) IAM's BLKSIZE will be 13682 - 4 blocks per track CISZ(20480) IAM's BLKSIZE will be 27998 - 2 blocks per track
DATACLASS(dataclas)	For DFSMS installations, this parameter specifies the name of the SMS DATA CLASS construct, which provides the allocation attributes for the new dataset. The DATACLAS name must have been previously established by the Storage Administrator. Attributes from the DATACLAS will be used, unless otherwise explicitly specified on the DEFINE statement. If the DATACLAS name contains the literal $IAM, and the dataset is SMS managed, then the dataset will be defined as an IAM dataset. Values provided by a Data Class include type of dataset, ESDS or KSDS, maximum record size, key length and key offset, space allocation values, free space, share options, CI size, and volume count.

10.20 CONTINUED . . .

FILE(ddname) Optional keyword that specifies a DD name that allocates the volume(s) on which the IAM file is to be allocated.

For IAM files, the primary use of this parameter is to provide a mechanism for relating IAM overrides, which are specified on a DDNAME level, with the specific file being defined. For example, if you specify an IAM CREATE Override with a DDNAME=FILE1, then to relate those overrides to the desired file, specify FILE(FILE1) on the DEFINE CLUSTER command. With IAM, there is no need for there to actually be a DD card in the step defining the file with that DDNAME. An alternative to using the file parameter is to code the CREATE override specifying the DSN= (dataset name) parameter.

If you are defining only one file in a particular job step, or want the same override for all of the files being defined, on the IAM Create Override statement specify DD=&ALLDD, and do not provide any FILE parameter on the define.

FREESPACE(CI%, CA%) For KSDS type of files, specifies the amount of space to be reserved for future inserts or updates when the file is being loaded.

CI%: Specifies the amount, as a percentage, of space to be left available in each prime block of the IAM file. For IAM, this value is used to calculate an Integrated Overflow percent. IAM's concept of Integrated Overflow is essentially the same as VSAM. IAM takes the value coded and leaves this percentage of each block empty, to accommodate adds (inserts) to the block. Unless a file is never updated and never has records added to it, some CI free space should be specified. This is of particular importance to data compressed files that are updated, because even if the application does not change the length, the stored record may end up being longer after compression.

CA%: For Enhanced format IAM datasets, this controls how much DASD space is released at the end of a file load. Using _ of the specified percentage, a target amount of DASD space to be reserved for future expansion is computed. If the amount of available DASD space within the file extent(s) is equal to or less than the amount to be reserved, then no space is released. IAM will not go after additional extents to meet the space reservation. If the amount of DASD space exceeds the reserved value, then the excess will be released.

Users of compatible format IAM datasets should review the section on considerations for compatible mode files for additional information on the implications on the file structure of this parameter.

KEYRANGES(low high) IAM will not split the data into key ranges. However if specified, IAM will make the secondary allocation quantity equal to the primary. This is because VSAM would use the primary for each key range, which IAM is not able to do.

LINEAR Specifies that a LINEAR type of CLUSTER that is being defined. IAM does not support this type of VSAM dataset. Specification of LINEAR for an IAM dataset will result in the DEFINE failing.

10.20 CONTINUED . . .

LOG(NONE|UNDO|ALL) For IAM files that are DFSMS managed, this specifies the recoverability of a dataset, and the journaling that IAM will perform. Specification of this parameter will have a different impact on IAM than with VSAM, because IAM provides journaling capabilities that VSAM does not. (Alternatively, journaling options can be specified on the IAM CREATE override JRNAD=). Specifying NONE indicates that the file is not recoverable and IAM will not perform any journaling. Specifying UNDO indicates that the file is recoverable, and IAM will journal the information necessary for backing out updates. Specifying ALL indicates that the file is recoverable and IAM will journal the information necessary for either a backout recovery, or a forward recovery. Refer to the section on IAM Journaling, 10.88, for additional information on IAM journal and recovery services.

MGMTCLASS(management class name) For SMS installations, this parameter specifies the name of the SMS Management Class for the new dataset.

MODEL(datasetname) Specifies that the attributies of the dataset being defined will be copied from an existing VSAM or IAM dataset. This capability is only relevant for basic file characteristics, such as record size, key length and offset, space allocation values, volumes, and free space values. Any IAM overrides are NOT picked up by the MODEL parameter. Likewise, any VSAM file attributes that are ignored by IAM are not available either, such as IMBED, SPEED, REPLICATE, KEYRANGES, etc.

If you are trying to define a VSAM file using an IAM file as a model, you MUST provide an OWNER parameter, with a value that does not contain $IAM.

NOTE: When using MODEL with ANYVOL, the SUBALLOCATION parameter must also be specified. Although it is ignored by IAM, it prevents IDCAMS from trying to allocate ANYVOL.

NUMBERED Specifies that an RRDS type of CLUSTER that is being defined. IAM does not support this type of VSAM cluster. Specification of NUMBERED for an IAM dataset will result in the DEFINE failing.

RECATALOG s an optional keyword for existing IAM datasets to reestablish the catalog entry. Requires the user to specify the dataset name, the volume(s), and OWNER($IAM) if $IAM is not in the dataset name. Recatalog is also used after renaming one or more components of an alternate index sphere, to reset the dataset relation names to the new names.

REUSE
NOREUSE Specifies whether the file being defined can be reloaded (or reorganized) without being redefined. IAM defaults to REUSE, which is that any IAM file can be reloaded without having to be deleted and redefined. To use this feature with IDCAMS REPRO, specify the REUSE keyword.

IAM does provide a Global Option, ENABLE=NOREUSE, which if set will cause IAM to honor the specification of REUSE or NOREUSE. If that Global Option has been set, then IAM will honor the NOREUSE setting just like VSAM. While quite rare, there are a few application programs that rely on the NOREUSE setting.

NOREUSE will not allow a file to be reloaded without being deleted and redefined. An exception to this is made if the program issuing the OPEN is FDRREORG, in which case it will be allowed. If any other attempt is made to do so, the OPEN will fail with a return code of 8, and the ACB error flag set to 232, or x'E8'.

10.20 CONTINUED . . .

SHAREOPTIONS (cross-region ,cross-system) Specifies the level of protection provided by the access method to prevent or allow sharing of data within the file. The protection mechanisms include the OS/390 ENQ service, and the internal IAM buffering techniques.

The first parameter specifies how a file can be shared in the same system (CPU). The second parameter specifies how a file is shared between systems.

NOTE: With IAM RLS, IAM supports automatic record level sharing within the same MVS image for concurrent users of the IAM dataset. This is an enhanced sharing facility, that by default may be automatically provided for cross region share options values of 3 or 4. When IAM RLS is not active, IAM supports the cross-region share options with the MVS ENQ service, the same as VSAM. IAM does not support the cross-system share options. IAM issues an ENQ with a major name (QNAME) of IAMENQ and the dataset name plus first volume serial as the minor name (RNAME). If you need to enforce ENQ protection cross-system then you must add the major name of IAMENQ to your CA-MIM or IBM's GRS control files or whatever ENQ control product you use.

Cross Region Share Option Values:

1. Any number of users for read **OR** one user for update. The file's structure, data integrity, and read integrity are fully preserved.
2. Any number of users for read **AND** one user for update. The file's structure and data integrity are fully preserved. If the file is currently opened for update, other users reading the file do not have read integrity. They may not be able to access inserted or updated records, if such records were added to the overflow areas of the file, without closing and reopening the dataset.
3. Any number of users for read or update and users are responsible for integrity. Updated blocks are immediately written back out to DASD. *Use of this share option for IAM files is strongly discouraged, unless you share the dataset with IAMRLS*, or some other VSAM sharing software. Due to the nature and structure of the index to the IAM overflow area, the data integrity of IAM files is compromised by use of this share option value.
4. Any number of users for read or update, and users are responsible for integrity. IAM will use only a single buffer, and each logical I/O request will cause the buffer to be refreshed, and subsequently rewritten if the record is updated. *Use of this share option for IAM files is strongly discouraged, unless you share the dataset with IAMRLS*, or some other VSAM sharing software. Due to the nature and structure of the index to the IAM overflow area, the data integrity of IAM files is compromised by use of this share option value.

SPANNED IAM supports SPANNED records on Enhanced Format files, with a maximum record size of almost 8 megabytes.

STORCLASS(Storage class name) For SMS installations, this parameter specifies the name of the SMS Storage Class construct. For datasets that are to be placed on SMS volumes, Storage Class must be either implicitly specified by the ACS routines, or explicitly specified on the DEFINE command. If the Storage Class Name contains the literal $IAM, the file will be defined as an IAM file.

SUBALLOCATION IAM files are always allocated as if they were unique clusters. However, it may be necessary to specify this parameter when using the IAM non-specific allocation (ANYVOL) and the MODEL parameter. SUBALLOCATION will prevent IDCAMS from allocating the volumes indicated in the VOLUME parameter when the MODEL parameter is specified.

10.20 CONTINUED . . .

TO(date)
FOR(days)
Specifies the retention period for the file being defined. This parameter has the same meaning for an IAM file as a VSAM file. The expiration date is placed in the VTOC for the dataset, and in the catalog entry. The keyword PURGE must be specified on the DELETE to cause the file to be scratched.

TO(date) – gives the date in the form YYYYDDD (four or two digit year and three digit Julian date), through which the IAM file defined is to be date protected.

FOR(days) – gives the number of days up to 9998, through which the IAM file being defined is to be date protected. A value of 9999 results in permanent retention.

Default: Dataset is not date protected.

UNIQUE
This parameter has no relevance for IAM files, as IAM files are always unique.

The user should be aware that specification of this keyword results in IDCAMS allocating the specified volumes prior to issuing the actual define request. For this reason, it is recommended that this parameter not be specified for IAM files.

If the customer is using the IAM non-specific device allocation, i.e. ANYVOL, then this parameter must not be specified.

EXAMPLES OF DEFINING IAM DATASETS WITH IDCAMS

The following is a set of examples demonstrating how to define IAM datasets using IDCAMS. The first example demonstrates how to convert a DEFINE for a VSAM dataset to an IAM dataset. The subsequent examples demonstrate various different ways of using IDCAMS to define IAM datasets. All of the included examples have a LISTCAT after the define. When a LISTCAT ALL is done for an IAM dataset, IAM will dynamically allocate an IAMPRINT DD to SYSOUT which will contain detailed information on the IAM dataset. The output from IDCAMS itself will indicate that the IAM file is a non-VSAM dataset.

Note: The JCL used for the examples through out the Users Guide are contained in the JCL library on the IAM installation tape. Check with the person who installed IAM to see if that library is available to you.

10.20 CONTINUED . . .

EXAMPLE A: CONVERSION OF VSAM DEFINE TO IAM DEFINE

This side by side example demonstrates how simple it is to convert a VSAM cluster definition to an IAM dataset definition. The only change required was adding the parameter OWNER($IAM) under the CLUSTER level of the Define control statement. For many VSAM KSDS or ESDS types of files, this is all that is necessary to convert the file to IAM. Starting with the next file load, all of the performance advantages and features of IAM are available for this dataset.

```
Original VSAM Define                          IAM DEFINE

//DEFINE    EXEC PGM=IDCAMS                   //DEFINE   EXEC PGM=IDCAMS
///SYSPRINT  DD SYSOUT=*                      //SYSPRINT  DD SYSOUT=*
//SYSIN      DD *                             //SYSIN     DD *
  DEFINE CLUSTER                   -            DEFINE CLUSTER                   -
    (NAME(EXAMPLE1.DATASET)        -              (NAME(EXAMPLE1.DATASET)        -
    VOLUMES(VOL001)                -              VOLUME(VOL001)                 -
    CYL(10 1)                      -              CYL(10 1)                      -
                                          ADD → OWNER($IAM)                      - ← ADD
    SPEED REUSE )                  -              SPEED REUSE )                  -
    DATA(                          -              DATA(                          -
    NAME(EXAMPLE.DATASET.DATA)     -              NAME(EXAMPLE.DATASET.DATA)     -
    RECORDSIZE(200 256)            -              RECORDSIZE(200 256)            -
    KEYS(16 0)                     -              KEYS(16 0)                     -
    CISZ(4096)                     -              CISZ(4096)                     -
    FREESPACE(10 10) )             -              FREESPACE(10 10) )             -
    INDEX(                         -              INDEX(                         -
    NAME(EXAMPLE.DATASET.INDEX)-                  NAME(EXAMPLE.DATASET.INDEX)-
    CISZ(1024) IMBED) )                           CISZ(1024) IMBED) )
  LISTCAT ENT(EXAMPLE1.DATASET) ALL             LISTCAT ENT(EXAMPLE1.DATASET) ALL
/*                                            /*
```

Figure 1: Example Conversion of Define from VSAM to IAM (EX1020A)

EXAMPLE B: BASIC IAM KSDS DEFINE

This example demonstrates a basic IAM dataset definition. Note, in comparison to the above example, that the DATA and INDEX component sections are eliminated. There is no need for them to exist for IAM dataset, as IAM consists of a single physical dataset, which uses the name specified for the CLUSTER as it's dataset name. This example includes the optional parameters of FREESPACE, SHAREOPTIONS, and REUSE. While in general REUSE is not necessary, as it is the default for IAM datasets, it is specified just in case the installation has changed the IAM Global Options to ENABLE=NOREUSE, in which case the IDCAMS default of NOREUSE takes effect without the explicit specification of REUSE.

```
//IAMDEFIN  EXEC PGM=IDCAMS
//SYSPRINT   DD  SYSOUT=*
//SYSIN      DD  *
  DEFINE CLUSTER           -
    (NAME(MY.IAM.KSD)      -
    OWNER($IAM)            -
    VOLUMES(MYVOL1)        -
    CYL(10 2)              -
    RECORDSIZE(100 1000)   -
    KEYS(24 8)             -
    FREESPACE(5 20)        -
    SHAREOPTIONS(2 3)      -
    REUSE  )
   LISTCAT ENT(MY.IAM.KSD) ALL
/*
```

Figure 2: Example Basic IAM KSDS Definition (EX1020B)

10.20 CONTINUED . . .

EXAMPLE C: BASIC IAM ESDS FILE DEFINE WITH $IAM IN DATASET NAME

This example demonstrates a basic ESDS dataset definition. The file will be defined as an IAM dataset due to the literal '$IAM' included in the dataset name. Also note that the keyword NONINDEXED has been included in the define to indicate that the file is to be an ESDS type of file.

```
//DEFINESD EXEC PGM=IDCAMS
 //SYSPRINT DD   SYSOUT=*
 //SYSIN    DD   *
  DEFINE CLUSTER               -
     (NAME(MY.ESDS$IAM.FILE)   -
       VOLUMES(MYVOL1)         -
       CYL(200 50)             -
       RECORDSIZE(80 100       -
       CISZ(4096)              -
       SHAREOPTIONS(2 3)       -
       NONINDEXED              -
   LISTCAT ENT(MY.ESDS$IAM.FILE) ALL
 /*
```

Figure 3: Basic IAM ESDS Define (EX1020C)

EXAMPLE D: DEFINE OF AN IAM ESDS FOR > 4 GIGABYTE

IAM ESDS datasets can exceed 4 gigabytes, providing that the application has no dependency on the RBA's (Relative Byte Addresses) of each record being the identical value as VSAM's. To use this capability specify the IAM CREATE Override keyword PSEUDORBA. This indicates to IAM that it can generate a 4-byte RBA that is different than the one VSAM would use for the same record. In effect, IAM will be returning a 4 byte relative record number. Amongst the applications that cannot use this feature is SAP, which is dependent upon the normal VSAM RBA values being returned.

```
//BIGESDS  EXEC PGM=IDCAMS
//SYSPRINT  DD  SYSOUT=*
//IAMOVRID  DD  *
  CREATE    DD=&ALLDD,PSEUDORBA
/*
//SYSIN     DD   *
  DEFINE CLUSTER        -
    (NAME(MY.BIG.ESDS)  -
    OWNER($IAM)         -
    CYL(1100 1100)      -
    NONINDEXED          -
    RECORDSIZE(200 4089)-
    CISZ(4096)          -
    VOLUMES(V33901 V33902 V33903 V33904)  )
  LISTCAT ENT(MY.BIG.ESDS) ALL
/*
```

Figure 4: Define of > 4 GIGABYTE IAM ESDS File (EX1020D)

10.20 CONTINUED . . .

EXAMPLE E: DEFINE AN IAM DATASET USING A MODEL DATASET

In the example below, an IAM dataset is defined using a model dataset. The model dataset can be either VSAM or IAM. Amongst the attributes that will be used from the model, unless otherwise explicitly specified on the define request, are file format (KSDS or ESDS), record sizes, key length and offset, free space values, volume, and control interval size.

```
//DEFMODEL EXEC PGM=IDCAMS
//SYSPRINT  DD  SYSOUT=*
//SYSIN     DD  *
  DEFINE CLUSTER                    -
    (NAME(MY.NEW.IAMFILE)           -
    OWNER($IAM)                     -
    MODEL(MY.INDEXED.FILE) )
  LISTCAT ENT(MY.NEW.IAMFILE)
/*
```

Figure 5: Example of an IAM Define with Model (EX1020E)

EXAMPLE F: DEFINE OF AN SMS MANAGED IAM DATASET

In the example below, an IAM dataset is being defined that will be SMS managed. The SMS classes are explicitly specified by the user. One or more of the SMS classes could be automatically selected by the ACS routines. All of the dataset attributes are being determined by SMS from the appropriate SMS classes. The following attributes are taken from the specified Data Class:

- Type of dataset (KSDS or ESDS)
- Maximum Record Size (LRECL)
- Key length and offset (for KSDS datasets)
- Free Space and Share Options
- Space allocation parameters
- Control Interval Size

The dataset will become an IAM dataset because the $IAM literal is contained within the Data Class name. Note that while a Data Class can be specified for datasets not managed by SMS, for IAM to pick up the Data Class name the dataset MUST be SMS managed. Otherwise, IDCAMS does not pass the Data Class value on the DEFINE, although the other Data Class attributes are included.

```
//DEFSMSDS EXEC PGM=IDCAMS
//SYSPRINT  DD  SYSOUT=*
//SYSIN     DD  *
  DEFINE CLUSTER           -
    (NAME(MY.SMS.FILE)     -
    DATACLAS(MY$IAM1)      -
    MGMTCLAS(DBSTNDRD)     -
    STORCLAS(BASE) )
  LISTCAT ENT(MY.SMS.FILE) ALL
/*
```

Figure 6: Define of an SMS Managed IAM Dataset (EX1020F)

10.20 CONTINUED . . .

EXAMPLE G: DEFINING MULTIPLE IAM DATASETS WITH OVERRIDES

This is an example of defining multiple IAM datasets within the same IDCAMS job step. Each IAM dataset has it's own unique overrides. Each of the overrides identifies the related dataset by using the DSN= (dataset name) operand. The first dataset has very high I/O activity, so it is being defined with a MAXBUFNO override to increase buffering. The second dataset is used by COBOL programs with a variable length record lay out giving a maximum record size of 32,200. It is defined as containing SPANNED so that it will not use an inefficient block size of 32760. This dataset has an override to indicate that it is to use hardware compression.

```
//DEF2FILE EXEC PGM=IDCAMS
//SYSPRINT  DD  SYSOUT=*
//IAMOVRID  DD  *
  CREATE  DSN=MY.HEAVYIO.CLUSTER,MAXBUFNO=512
  CREATE  DSN=MY.BIGREC.CLUSTER,DATACOMP=HW
/*
//SYSIN     DD  *
  DEFINE CLUSTER                  -
     (NAME(MY.HEAVYIO.CLUSTER)-
     OWNER($IAM)                  -
     RECORDSIZE(1000 2000)        -
     KEYS(24 0)                   -
     VOLUMES(MYVOL1 MYVOL2)       -
     CYL(500 100)                 -
     FREESPACE(10 10)             -
     SHAREOPTIONS(2 3)            -
     REUSE  )
  LISTCAT ENT(MY.HEAVYIO.CLUSTER) ALL
  DEFINE CLUSTER                  -
    (NAME(MY.BIGREC.CLUSTER)      -
    OWNER($IAM)                   -
    RECORDSIZE(500 32000)         -
    KEYS(32 0)                    -
    VOLUMES(MYVOL3 MYVOL4)        -
    CYL(1000 200)                 -
    FREESPACE(10 10)              -
    SHAREOPTIONS(2 3)             -
    CISZ(8192)                    -
    REUSE  SPANNED )
  LISTCAT ENT(MY.BIGREC.CLUSTER) ALL
/*
```

Figure 7: Multiple Dataset Define (EX1020G)

10.20 CONTINUED . . .

DEFINING IAM FILES IN JCL

JCL ALLOCATION IAM datasets can be defined in JCL, just as VSAM datasets can. DFSMS must be active within the system to use this facility, however the dataset does not have to be DFSMS managed. Similar to VSAM, only a subset of the attributes can be specified in JCL. Use of a DATACLAS is encouraged, because that will pick up the free space and share option values, which cannot be specified in JCL. Also, because the JCL defines are actually done under the initiator TCB, IAM does not have access to any IAM Overrides that may be included in the JCL for the define process. The overrides can be specified on the job step that loads the dataset, which may or may not be the step that allocates the IAM dataset. Because there is no mechanism to pass an OWNER parameter, the dataset must either have the literal '$IAM' within the dataset name, or be a DFSMS managed dataset with '$IAM' in either the STORCLAS or DATACLAS name.

IAM files can also be allocated as temporary datasets, with the same restrictions as temporary VSAM datasets. The dataset will not be cataloged, and it is restricted to a single volume.

To use the IAM JCL define capability, IAM MUST BE IN LINKLIST! Because the definition is running under the initiator TCB, there is no access to the STEPLIB or JOBLIB to do the allocation. If you have multiple levels of the IAM active, then the one in the link list will be then one to perform the define.

10.20 CONTINUED . . .

JCL KEYWORDS FOR IAM DATASET DEFINITION

The following is a list, with a brief description of the JCL DD card keywords that are used when allocating an IAM dataset through JCL.

KEYWORD	DESCRIPTION
DATACLAS=	Specifies the name of the DFSMS Data Class being requested for this dataset. The file attributes will be obtained from this class. For DFSMS managed datasets, if the name includes the literal $IAM, the dataset will be an IAM dataset.
DSN=	Specifies the 1 to 44 character dataset name. For IAM datasets that are not DFSMS managed, the literal $IAM must be part of the dataset name. Temporary datasets begin with a single or double &.
DISP=	Specifies the disposition of the dataset. All VSAM and IAM datasets, except temporary datasets, must be cataloged, and will be cataloged during step initiation when the dataset is allocated. Valid values include: (NEW,CATLG) for permanent datasets, or (NEW,PASS) for temporary datasets.
KEYLEN=	For KSDS type of files, specifies the length of the key. This will override any value determined from a DATACLAS or MODEL if specified. For IAM, valid values are 1 to 249.
KEYOFF=	For KSDS type of files, specifies the offset of the key within the record (RKP). For IAM files, this must be less than or equal to 4091.
LIKE	Specifies the name of an existing IAM or VSAM dataset from which the RECORG, KEYLEN, KEYOFF, LRECL, and space attributes will be obtained.
LRECL=	Specifies the maximum record length. For KSDS type of files, this must be at least the value of KEYLEN + KEYOFF.
MGMTCLAS=	Specifies the DFSMS Management Class that is being requested for this dataset.
RECORG= [KS | ES]	Indicates the type of dataset to define. Valid values for IAM are KS for a KSDS or ES for an ESDS. IAM does not support the RR or LS type of VSAM datasets.
STORCLAS=	Specifies the DFSMS Storage Class that is being requested for the dataset.
UNIT=	Specifies the type of device to which the dataset is being allocated.
VOL=SER=	Specifies the volume(s) to which the dataset is being allocated. Note that temporary datasets are limited to one volume. This is not required for DFSMS managed datasets.

EXAMPLES OF DEFINING IAM DATASETS WITH JCL

The following are some examples of how to define IAM files through JCL. For clarity in the examples, the program being executed is IEFBR14. However, any program could be executed, including programs that may load and access the dataset.

10.20 CONTINUED . . .

EXAMPLE H: BASIC JCL DEFINE OF AN IAM FILE

The example below is a basic example of the define of an IAM KSDS type of file through JCL. The dataset eligibility for an SMS managed volume will be determined by the installations ACS routines.

```
//JCLDEFIN EXEC PGM=IEFBR14
//NEWIAMFL  DD  DSN=new.my$IAM.file,DISP=(NEW,CATLG),
//     RECORG=KS,KEYLEN=8,KEYOFF=0,LRECL=128,
//     UNIT=SYSDA,VOL=SER=MYVOL1,
//     SPACE=(CYL,(20,2))
```

Figure 8: Basic JCL Define of an IAM File (EX1020H)

EXAMPLE I: DEFINE OF AN IAM DATASET WITH SMS CLASSES

In the following example, the file attributes are picked up from the DATACLAS, as well as the request to make the file an IAM file.

```
//SMSDEFIN EXEC PGM=IEFBR14
//NEWIAMFL  DD  DSN=new.myfile.cluster,DISP=(,CATLG),
//    DATACLAS=(MY$IAMF1),STORCLAS=(PERMVSAM)
```

Figure 9: JCL Define of an SMS Managed IAM Dataset (EX1020I)

EXAMPLE J: DEFINE OF A TEMPORARY IAM DATASET

The example below allocates a temporary IAM dataset, whose attributes are being taken from an existing IAM or VSAM dataset.

```
//TEMPIAM  EXEC PGM=IEFBR14
//TEMPFILE  DD  DSN=&&TEMP.$IAMFIL,DISP=(,PASS),
//     LIKE=my.indexed.cluster,
//     UNIT=SYSDA
```

Figure 10: JCL Define of Temporary IAM Dataset (EX1020J)

EXAMPLE K: DEFINE OF AN IAM ESDS DATASET

In the following example, an IAM ESDS type of file is being defined.

```
//ESDSDEF1 EXEC PGM=IEFBR14
//ESDSFILE  DD  DSN=my.esds$IAM.cluster,DISP=(,CATLG),
//     RECORG=ES,LRECL=1020,
//     UNIT=SYSDA,VOL=SER=MYVOL1,SPACE=(CYL,(2,1))
```

Figure 11: JCL Example Define of an IAM ESDS File (EX1020K)

10.20 CONTINUED . . .

DEFINING IAM DATASETS UNDER TSO

TSO DEFINES

IAM datasets can be defined under TSO, using a variety of methods. TSO offers an IDCAMS Define process through the DEFINE CLUSTER command. TSO also has a method similar to the JCL define, by using the TSO ALLOC command. IAM provides a set of ISPF panels, and the underlying software, which includes the capability to define an IAM dataset through an easy to use fill in the blanks method. Many other products offering various capabilities under ISPF for VSAM files will also work with IAM files.

The one major difference under TSO is that all of the above methods utilize the standard TSO dataset naming conventions. If the dataset name is not specified within apostrophes, then the dataset name will be prefixed. This prefix is normally your TSO user id, but can be changed.

DEFINING IAM DATASETS WITH THE IAM ISPF PANELS

One of the easiest ways to allocate an IAM dataset is through the IAM ISPF panels. They feature a fill in the blanks mechanism, along with providing easy specification of various IAM overrides all on one screen. To get to the IAM Define panel, select option I on the IAM Primary Option Menu. Be sure to fill in the dataset name field and the dataset type field. Optionally, you can specify an existing IAM or VSAM dataset as a model for the dataset attributes. Below is an example of the IAM Primary Option Menu with the user provided text highlighted

IAM ISPF PRIMARY OPTION MENU

```
-------------------------- IAM PRIMARY OPTION MENU --------------------------
OPTION ===> I

  I     - Allocate (DEFINE) a new IAM Dataset
  V     - Allocate (DEFINE) a new VSAM Cluster
  D     - Delete a Dataset, Cluster, Path, or Alternate Index
  C     - Copy an IAM Dataset, VSAM KSDS, or VSAM ESDS
  M     - Move an IAM Dataset, VSAM KSDS, or VSAM ESDS
  R     - Rename a Dataset, Cluster, Path, or Alternate Index
  U     - Invoke an IAM Utility program
blank - Display IAM Dataset or VSAM Cluster Information


Enter dataset name (required for all options except U)
Dataset Name       ===>  myiam.cluster
Dataset Type       ===> C  C=Cluster X=AIX P=Path (options I, V Only
Enter Model or New dsname (optional for options I and V, required for option R)
Model|Newname      ===>


Delete Confirmation ===> YES  Yes|No
```

Figure 12: IAM Primary Options ISPF Panel

Once you hit enter, the IAM Definition panel is displayed. If you had specified a MODEL dataset, the information from the model is filled in on the definition panel, any of which can be changed. This panel includes the various parameters and attributes that can be specified for the file. Note that this panel displays the fully qualified dataset name at the top left. The attributes essentially match those that must be specified on an IDCAMS Define Cluster request. Additionally, on the right side of the panel is an area to specify the various IAM Overrides that affect the file definition. Once the necessary attributes are specified, then hit enter to allocate the file. Note that the data entered on the panel is highlighted text in the example.

10.20 CONTINUED . . .

IAM ISPF DEFINE PANEL

```
----------------------------- DEFINE AN IAM FILE -----------------------------
COMMAND ===>

Dataset Name: RAM2.MYIAM.CLUSTER
                                       Multi-Volume Allocation ===> NO
ALLOCATION
 Volume             ===> junk01   IAM OVERRIDES
 SMS Storage Class ===>                ANYVOL Unit       ===>
 SMS Data Class     ===>                Blocking Factor   ===>          1-15,>300
 SMS Mgmt Class     ===>                Overflow Records ===>          0-2000000
 Cyls|Recs|Trks     ===> cyls           Var. Overflow     ===> YES      Yes|No
 Primary Space      ===> 10             Prime Extension   ===>          0-32767
 Secondary Space    ===> 2              Space Release     ===>          Yes|No
 Recatalog          ===> NO             Data Compress     ===>          Yes|No

ATTRIBUTES                              Enhanced Format   ===> YES      Yes|No
 KSDS|ESDS          ===> KSDS           Minbufno          ===>          1-32
 Max Recordsize     ===> 256            Maxbufno          ===>          1-32
 Avg Recordsize     ===> 100            PSEUDORBA (ESDS) ===>          Yes|No
 Key Length         ===> 8              Extended ESDS     ===>          Yes|No
 Key Offset         ===> 8
 CI Size            ===> 4096           RETENTION
 CI/CA Free %       ===> 10 / 10        DAYS              ===>          0-9999
 Shareoption        ===> 2              EXPIRATION DATE   ===>          YYYY.DDD
```

Figure 13: IAM Dataset Define Panel

If all the required parameters have been specified, and the allocation is successful, then the IAM Primary Option Menu will be redisplayed, with the status message in the top right corner. Notice the highlighted feedback, which indicates that the IAM dataset was successfully allocated. To verify that the file has been define how you want it, press enter to have the IAM dataset attribute ISPF panel displayed.

```
-------------------------- IAM PRIMARY OPTION MENU -------- DATASET ALLOCATED
OPTION ===>

I     - Allocate (DEFINE) a new IAM Dataset
V     - Allocate (DEFINE) a new VSAM Cluster
D     - Delete a Dataset, Cluster, Path, or Alternate Index
C     - Copy an IAM Dataset, VSAM KSDS, or VSAM ESDS
M     - Move an IAM Dataset, VSAM KSDS, or VSAM ESDS
R     - Rename a Dataset, Cluster, Path, or Alternate Index
U     - Invoke an IAM Utility program
blank - Display IAM Dataset or VSAM Cluster Information

Enter dataset name (required for all options except U)
 Dataset Name       ===> MYIAM.CLUSTER
 Dataset Type       ===> C  C=Cluster X=AIX P=Path (options I, V Only)

Enter Model or New dsname (optional for options I and V, required for option R)
Model|Newname      ===>

Delete Confirmation ===> YES  Yes|No
```

Figure 14: IAM Dataset Define Confirmation

10.20 CONTINUED . . .

IAM FILE CHARACTERISTICS DISPLAY

IAM responds with the IAM File Characteristics display, with the various attributes filled in. The format of the display is based on the format of the file being displayed. There are displays for IAM Enhanced format files, as seen below, a display for IAM Compatible format files, and a display for VSAM files.

```
------------------------- IAM FILE CHARACTERISTICS -------------------------
COMMAND ===>
Dataset Name: RAM2.MYIAM.CLUSTER

Definition                          Allocation
  Record Length:        256           Volume:             JUNK01
  Record Format:  VARIABLE            Device Type:          3380
  Key Length:             8           Tracks in use:           0
  Key Offset:             8           Block Size:           4096
  Dataset Type:        KSDS           Blocking Factor:      4096
  Share Option:           2           Alloc Type:      CYLINDERS
  Release:              YES           Primary Alloc:          10
  Storage Class:                      Secondary Alloc:         2
  Data Class:                         Compressed Keys:        NO
  Mgmt Class:                         Compressed Data:        NO

Statistics                          Extended Area
  Creation:        2000.329           Overflow Records:          0
  Expiration:      0000.000           Blocks Allocated:          0
  Last Reference:  2000.329           Blocks Used:               0
  Records:                0           Overflow Blocks:           0
  Deletes:                0           PE Blocks:                 0
  Inserts:                0           Blocks Available:          0
  Updates:                0           Variable Overflow:       YES
  Minimum Buffers:
  Maximum Buffers:
  Extended File Fmt:    YES

Associations
```

Figure 15: IAM ISPF Dataset Attributes Display

DEFINING IAM DATASETS WITH THE TSO DEFINE CLUSTER COMMAND

IAM datasets can also be defined with the TSO Define Cluster command. While this does not provide the same ease of use as the IAM ISPF panels, it is an option if the IAM ISPF panels have not been installed. This command essentially is identical to the IDCAMS Define Cluster command, with the same format and parameters. Some of the abbreviations may be slightly different. Refer to the IDCAMS Define section for information relating to the parameters required for IAM datasets. Also, you can issue the TSO command: "HELP DEFCL" to get the information about the TSO DEFINE CLUSTER command.

Remember that the dataset name fields use the TSO naming conventions. The names will be prefixed unless enclosed within apostrophes.

10.20 CONTINUED . . .

EXAMPLE OF TSO DEFINE CLUSTER

```
Menu  List  Mode  Functions  Utilities  Help
------------------------------------------------------------------------------
                              ISPF Command Shell
ISPF Command ===>
Enter TSO or Workstation commands below:
===> define cluster(name(myiam.cluster) owner($iam) keys(8 0) recordsize(100
256) cylinders(10 1) vol(scr083) freespace(10 10) shareoptions(2 3) reuse)
Place cursor on choice and press enter to Retrieve command
=> help defcl
=>
```

Figure 16: Example of a TSO Define Cluster Command for an IAM Dataset

TSO responds with the following messages:

```
IDC0508I DATA ALLOCATION STATUS FOR VOLUME SCR083 IS 0
IDC0509I INDEX ALLOCATION STATUS FOR VOLUME SCR083 IS 0
***
```

Figure 17: TSO Response to Define Cluster Command

After the file has been defined, you can use the TSO LISTCAT ALL command to see the file attributes. The following example shows how to issue the LISTCAT and the results.

```
Menu  List  Mode  Functions  Utilities  Help
------------------------------------------------------------------------------
                              ISPF Command Shell
ISPF Command ===>

Enter TSO or Workstation commands below:

===> listcat ent(myiam.cluster) all

Place cursor on choice and press enter to Retrieve command

=> define cluster(name(myiam.file) owner($iam) keys(8 0) recsz(100 256) vol(sc
=> help defcl
```

Figure 18: Example TSO LISTCAT Command

10.20 CONTINUED . . .

TSO LISTCAT ALL OUTPUT

As a result of the LISTCAT ALL, IAM produces what is called an IAMPRINT Report, which is written directly to the TSO terminal, which is followed by the standard IDCAMS SYSPRINT output. Refer to the section on IAM Reports for complete details on the IAMPRINT Report. Note that depending on your terminal, the output may require multiple screens, and spacing may appear slightly different.

```
IAM FILE ANALYSIS - DSN=RAM.MYIAM.CLUSTER
-------------------------------------------------------------------------
FILE FORMAT -- =  ENHANCED     -  FILE STATUS ----------- =  UNLOADED
RECORD SIZE -- =       256     -  FREESPACE - CI% ------- =        10
CI SIZE ------ =       512     -  FREESPACE - CA% ------- =        10
BLOCK FACTOR - =         0     -  EXTENDED OVERFLOW ----- =         0 BLOCKS
KEY SIZE ----- =         8     -  EXTENDED OVERFLOW USED  =         0 %
KEY OFFSET --- =         0     -  EXTENDED PE ----------- =         0 BLOCKS
FILE TYPE ---- =      KSDS     -  EXTENDED ALLOCATED ---- =         0 BLOCKS
DEVICE TYPE -- =      3380     -  EXTENDED AVAILABLE ---- =         0 BLOCKS
VOLUME COUNT - =         1     -  SPACE USED ------------ =       N/A TRACKS
 VOLSER ------- =    SCR083     -  SPACE ALLOCATED ------- =       150 TRACKS
TOTAL EXTENTS  =         1     -  TOTAL SPACE ALLOCATED - =       150 TRACKS
PRIMARY SPACE  =        10     -  SECONDARY SPACE ------- =         1 CYL
MULTIVOLUME -- =   PRIMARY     -  MAX SECONDARY --------- =        10 CYL
RELEASE ------ =       YES     -  SHARE OPTIONS --------- =         2
FILE DEFINED - =  2002.164     -  06/13/2002 -  1:41 PM - =  13:41:06
-----------------------------------------------------------------------------
IAM499 IAMLISTC(8.0/01P ) PROCESSING COMPLETED
NONVSAM ------- RAM.MYIAM.CLUSTER
         IN-CAT --- CATALOG.TSOUSER
        HISTORY
            DATASET-OWNER---.........     CREATION--------1997.238
            RELEASE----------------2      EXPIRATION------0000.000
          VOLUMES
       VOLSER------------SCR083     DEVTYPE------X'3010200E'     FSEQN--------0
ASSOCIATIONS--------(NULL)
     ATTRIBUTES
```

Figure 19: TSO LISTCAT Output

10.20 CONTINUED . . .

DEFINING IAM DATASETS WITH THE TSO ALLOC COMMAND

IAM files can also be defined with the TSO ALLOC Command. This is similar to defining IAM files through JCL. This command provides far fewer specifications than the DEFINE Command, allowing only the basic essentials to be specified. The ALLOC Command has the same parameters as added to the JCL DD card: RECORG, LRECL, KEYLEN, and KEYOFF. The use of a DATACLAS will add a few more fields, such as share options and free space. The highlighted text indicates the command and the operands. For complete information on the ALLOC command, use HELP ALLOC.

```
Menu  List  Mode  Functions  Utilities  Help
-----------------------------------------------------------------------------
                              ISPF Command Shell
ISPF Command ===>

Enter TSO or Workstation commands below:

===> alloc f(iamfile) da(my$iam.cluster) recorg(ks) lrecl(256) keylen(8)
keyoff(0)  cyl space(1 1) vol(scr083) unit(3380)

Place cursor on choice and press enter to Retrieve command

=> del myiam.cluster
=>
```

Figure 20: Example of TSO ALLOC Command

The ALLOC Command will either respond with error messages, or return with no messages at all, which indicates success. As with the prior DEFINE CLUSTER Command, you can issue a LISTCAT ALL just to validate the allocation.

10.20 CONTINUED . . .

SUMMARY OF DEFINING IAM FILES

SUMMARY As you can see, there are several methods available for defining IAM files. The most common method is to use the IDCAMS DEFINE CLUSTER command, however there are other choices. The other methods include through JCL DD cards, or through various mechanisms under TSO. The key ingredient is that you have to indicate that the file is to be an IAM file instead of a VSAM file. This is done by using OWNER($IAM), or by placing $IAM within the dataset name, or by using a DATACLAS or STORCLAS with $IAM in the class name. For many files, this is the only change required to use IAM instead of VSAM.

10.25 USING HARDWARE COMPRESSION

OVERVIEW

With IBM's Z-series processors, IBM has made significant performance improvements with their hardware compression instruction. This performance improvement has sparked new interest in using the IBM hardware compression capabilities. Therefore, IAM Version 8.0 introduces a hardware data compression option to use the IBM Hardware Compression instruction. The IAM use of hardware compression is specified by an IAM CREATE Override when an IAM file is defined or loaded of DATACOMP=HARDWARE (or HW). At this time, IAM is providing a single compression dictionary that is designed to handle datasets with English text and some numeric data, and is best suited for files with name and address data. Customers can also create their own compression dictionaries for IAM to use, and specify the dictionary name by the DICT= IAM Override. If customer interest in using the hardware compression with IAM files is strong, we will consider enhancing the capabilities being provided with IAM to include additional compression dictionaries and automation of dictionary selection.

IAM COMPRESSION

The IAM compression technique uses a highly optimized proprietary software algorithm to compress data by eliminating strings containing repetitive byte values. Additionally, the IAM technique has very low cost for strings of data that could not be compressed. This technique, along with IAM being able to make as full use as possible of the capacity of each track, provided significant space savings for many datasets. Also, because this technique requires no dictionary, it reduces virtual storage and DASD storage requirements. While other compression techniques could provide greater compression, such techniques generally also came with a high price tag in terms of CPU consumption. On the Z/Series processors, IBM has made significant performance improvements with their hardware compression instruction, dramatically reducing the CPU time requirements for hardware compression. Because of this improvement, IAM now has an option to utilize the IBM hardware compression instruction.

HARDWARE COMPRESSION

The IBM Hardware Compression algorithm relies on compression and corresponding decompression dictionaries. These dictionaries allow for compression and expansion of repeating data patterns within the records. A couple of difficulties are encountered with such an algorithm. First, for optimal compression, the data must be previewed to find the repeating data patterns within the dataset. Then from those repeating data patterns, the most frequently observed data patterns are converted into compression and decompression dictionaries. Also, one must insure that the dictionaries are safely stored, because if the decompression dictionary is lost, then the data cannot be decompressed thereby becoming useless. An alternative method of building or selecting a dictionary is to scan the first few records as they are being loaded, and select a generic dictionary that appears to offer best compression. Such generic dictionaries can provide decent amount of compression, although not the most compression possible for any given file, and eliminate the need to store complete dictionaries that can be retrieved when the dataset is processed.

IAM USE OF HARDWARE COMPRESSION

In this initial support for hardware compression, IAM offers it as an optional method to compress data. The IAM software compression algorithm is still fully supported, and remains the default compression technique. The hardware compression option can be selected on an individual file basis by specification on an IAM CREATE Override. When IAM uses hardware compression, customers can use the default compression dictionary, or can use their own compression dictionary. The IAM provided compression dictionary is suited for data consisting of predominately English text, with some numerical information. The dictionary is ideally suited for data files containing name and address information, along with other associated information. For files with different data characteristics, customers will find it advantageous to create a customized dictionary from a sampling of their data. If customers select to use their own customized compression dictionary, IAM will store that dictionary within the dataset itself when the dataset is loaded, to insure access to the data in case the dictionary is subsequently changed.

Depending on customer feedback, the IAM support of hardware compression may be enhanced in future versions or maintenance releases. Such support could include additional dictionaries being provided, and possibly a dynamic dictionary selection based on previewing the initial data written out to a file.

10.25 CONTINUED . . .

CREATING A COMPRESSION DICTIONARY

There are a few steps required to build compression dictionaries for your files. The procedure consists of the following steps:

1. Create the control statements required for executing the IBM REXX EXEC that will read the data, and generate the dictionary. Review the information in 'SYS1.SAMPLIB(CSRBDICT)', which includes detailed instructions on using the exec.
2. Create a sequential dataset containing the data you want to build a dictionary for, or a representative subset of the data.
3. Execute the CSRBDICT REXX EXEC, with the sequential dataset you created previously, and with the control information you've decided on.
4. Assemble and link the dictionary generated by the CSRBDICT REXX EXEC into load module format.

USING CSRBDICT

Using the CSRBDICT REXX exec can be a rather intimidating task, as there are many parameters that can be specified. Determining the best settings for many of the parameters may require multiple executions of CSRBDICT, varying the various parameters, and then reviewing the results. The exec can run for a long time, using lots of CPU time to come up with a dictionary. It is easiest to start with a very basic execution and testing the resulting dictionary. If you are satisfied with the amount of compression you obtain, then go with that. For many files, this basic approach can yield excellent results. If you are looking for more compression, or attempting to create a dictionary for multiple datasets, then you can get more involved with varying parameter settings, and providing a more detailed layout of your data records to CSRBDICT.

To get you started, a basic set of control card input is being provided in the IAM ICL (Installation Control Library). The member name is BDICTEX1, and is shown below. This example uses the basic IBM recommended parameters, and provides two field statements. The first field statement describes the data up to and including the key, which is going to be ignored by CSRBDICT because IAM will not attempt to compress that data. The second field statement is for the rest of the data in the record, which will actually be subject to compression. All you need to do is to alter the starting position of on the second field card to indicate the appropriate position of the first byte after the key within the data file for which the dictionary is being built. When selecting the values to use in the CSRBDICT "spec" file, for the "dicts" field you must specify either "AF ASM" or "AFD ASM" so it will generate a file containing assembler language representation of the dictionaries.

```
**The following is an example for building a 4k entry dictionary
**just using a basic pattern scan.  The first field card indiates to skip
**the data that is up to and including the key.  The second field card is
**for the rest of the data in the record.
**results maxnodes maxlevels msglevel stepping prperiod dicts
  r       40000    64        3        f 7 2 7  1000     afd asm
**colaps opt treedisp treehex treenode dupccs
  aam    opt x       h       n        x
**FLD col type dcenmen             INT  intspec
  FLD 1   ns
  FLD 15  sa
  FLD end
```

Figure 21: Example input to CSRBDICT REXX exec

10.25 CONTINUED . . .

USING CSRBDICT (Continued)

The next step is to obtain a representative sample of the data that is contained within the file that you are going to be creating the dictionary for. If the file size is relatively small, say under 50,000 records, you can probably use the entire file. However, if it is larger or if CSRBDICT is taking too long to run, you can either take a sample of the records in the file, or revise the "stepping" value in the BDICTEX1 member so that the entire input file will not be used. So, for example to reduce the amount of data scanned to say a little less than half the data (in this case 3/7), change the stepping values from f 7 2 7 to f 3 2 7.

For execution parameters, you must also specify format-1 sibling descriptors, which is done by specifying the value 1 for the "sdfmt" field. For dictionary size try starting with 4(K) entries. From our limited testing, we had best overall compression results with a maximum of 4K entries. Depending on your data patterns, you may find that a larger or smaller dictionary size will yield better results. The CSRBDICT process can take a long time to run, so be patient. Shown below is an example of the command to execute the CSRBDICT REXX exec.

```
ex 'sys1.samplib(csrbdict)' '4 1 eb ''my.test.data'' ( ''IAM.ICL(BDICTEX1)'''
```

Figure 22: Example of executing CSRBDCIT REXX exec

ASSEMBLE THE COMPRESSION DICTIONARY

After successfully running CSRBDICT, there will be several output files. There are two files of primary interest to the IAM dictionary build process, and they will have the suffixes of ACDICTs1 and AEDICTs1, where "s" will be the number of K entries in the dictionary, either 1, 2, 4, 8 or H for 512 entries. These two datasets will be assembled and linked into a load module that can be used by IAM for the compression and expansion (decompression) dictionary.

In the IAM ICL (Installation Control Library) is an example, HWDASM, of the JCL to assemble and link the dictionaries for use by IAM. The first step ASMACD assembles the compression dictionary. Change the name of the SYSIN dataset to the name that was created by CSRBDICT for your file, it will have the suffix ACDICTx1. The second step, ASMAED, assembles the expansion dictionary. Change the SYSIN dataset to the name that was created by CSRBDICT for the expansion dictionary for your file. It will have a suffix of AEDICTx1. The third and final step, LKED, will link the two dictionaries together into a load module that IAM can use. The first four characters of the dictionary name must be 'IAMD', and you can choose the last four characters. Make sure that the characters you use do not conflict with any existing load module names or other dictionaries. A recommendation is to set the first optional character to the "s" value of H, 1, 2, 4, or 8, and then select three other alphanumeric characters. It is recommended that you place it in a load module library other than the IAM library, so that way it will not be lost when a new level or version of IAM is installed.

USING THE COMPRESSION DICTIONARY

To use the dictionary that you just created, define an IAM file using an IAM Override card specifying hardware compression, and a dictionary name with the last four characters that you selected for the dictionary name. For example, if you chose 4ABC in the prior step, use the following override:

```
//IAMOVRID   DD   *
      CREATE      DD=&ALLDD,DATACOMP=HW,DICT=4ABC
/*
```

You can now load your file using your compression dictionary. After the file is loaded, if you perform a LISTCAT, the IAMPRINT report should indicate that the dataset is hardware compressed, with a dictionary name of 4ABC, and that the dictionary is stored in the IAM dataset.

After seeing the amount of space used, you may want to try changing some of the parameters for CSRBDICT to see if you can obtain better compression. If so, change the parameters and rerun the build process, making sure that you've noted your prior parameters and the results. Once you are happy with the results, you can save your final resulting dictionary.

10.30 LOADING IAM DATASETS

IAM FILE LOAD PROCESS

After successfully defining an IAM Dataset, the next step is to load data into the dataset. Generally, the file load process is where the dataset is populated with records, and the index structure is built. For IAM datasets, this portion of the index is called the prime index. Once the load is completed, the prime index for the dataset is established, and will not change. The load process is restrictive in that the only I/O operation allowed is to write new records into the dataset. For KSDS type of files, the records must be written in ascending key sequence. The file load must write one or more records to the dataset, and subsequently successfully close the dataset for it to be considered a loaded dataset. Once loaded, the dataset can be processed with the full range of I/O operations.

IAM datasets can be loaded by any program designed to load VSAM clusters. This includes system utility programs, such as IDCAMS REPRO and the SORT, as well as application programs. Each I/O request must be checked for successful completion, including the OPEN and the CLOSE. As will be explained in the subsequent paragraphs, critical file structure related processing occurs during Open and Close processing. If these processes do not complete successfully the dataset cannot be used. Failures during the file load process will generally result in the dataset having to be reloaded.

OPEN PROCESSING

When the IAM dataset is opened for the file load, IAM reads in the attributes, parameters and overrides that were specified when the file was defined, and checks for current overrides. Any IAM overrides provided for the file load will supersede the values specified when the file was defined. Using that merged information set, all of the file attributes are validated and established. Conflicts and errors within the merged attribute set will cause the OPEN to fail with a return code of 8, and an appropriate error code. Because most of the errors will be caught during the define process, file attribute errors are very rare during the open process. If one does occur, there will be an error message in the Job Log indicating the reason for the error. After the Open process successfully completes, the program can begin writing records into the dataset.

RECORD PROCESSING

As each record is written, IAM validates the length, and for KSDS type files does key sequence checking. A key sequence error is not considered by IAM to be a fatal error for a file load, although it may indicate a fatal error for the application. For ESDS type of files, IAM calculates an RBA value to be used as the internal key. The RBA value is based on the uncompressed size of the record, the RBA of the prior record, and the requested CI size. If the IAM data compression feature is enabled for this file, IAM will compress the record. As long as the compressed length is less than the uncompressed length, the compressed form of the record is written to the file. As each block is filled, leaving room for the Integrated Overflow (CI% free space), the block is scheduled to be written, and the highest key in each block is temporarily saved in either a data space, or in a dynamically allocated work file. Additional DASD space for the IAM dataset is acquired as necessary.

CLOSE PROCESSING

After all the data has been written to the file, the application program must explicitly Close the dataset. When the Close is issued, IAM will then read the through the temporarily saved high key structure, and format the index section of the file. When this process is successfully completed IAM returns to the calling program, and the dataset is now ready for use. Because errors can occur during this index build process, in particular insufficient DASD space to hold the index, it is important to check for a successful completion of the Close. An error during Close will result in a return code of 4, and an appropriate error code returned in the ACB.

10.30 CONTINUED . . .

BUFFERING The file load process is essentially a sequential output process. To optimize the I/O process, IAM uses a different buffering technique than the Real Time Tuning that is used on normal file access and update. During open processing, IAM acquires a pool of buffers to be used, based on the default or overridden value for CRBUFOPT. The buffer pool is split into two pieces. When the first set of buffers have been filled with data, IAM then issues the I/O for those buffers. Processing continues with the application passing records, which are placed into the second set of buffers. IAM is effectively providing I/O and processing overlap. When the second set of buffers is filled, IAM issues the I/O request to write all of those buffers, and waits if necessary for the I/O on the prior buffer set to complete. When that I/O completes, IAM continues accepting records into the first set of buffers. For efficient physical I/O, IAM always writes out data in full track increments. The largest number of buffers that will be used for a file load is specified by CRBUFOPT=MCYL. In that case, IAM acquires enough storage for two cylinders worth of data, and will write out a complete cylinder per physical I/O. The fastest file load is obtained by using CRBUFOPT=MCYL, which is the default setting from the IAM Global Options. Remember to supply buffers for the input file as well.

FILE FULL, OR SX37 ERROR A file full error during the load process is considered a fatal error by IAM. Because the index is written at the end of the dataset, an out of space condition will prevent the index from being properly written out to DASD. There will usually be messages indicating some type of Sx37 error condition. To insure that IAM is able to properly clean up after such a failure, in particular to release the ENQ that are issued, IAM will attempt to avoid an actual Sx37 abend under CICS. In that case, the request will fail with a file full logical error. A file full error will also be raised if the area being used for the temporary storage of the high keys is filled. This is considered a fatal error to the file load. Either increase the storage available for the data space, by using the DATASPACE override, or if not using the data space, increase the allocation values for the work file in the IAM Global Options Table, with the WORKPRIMARY and WORKSECONDARY values.

SINGLE RECORD LOADS SPECIAL CONSIDERATIONS Some applications use a technique for loading files that consists of loading a single record, followed by a mass insert. In some instances this technique is obvious due to an IDCAMS REPRO of a special record into the file. Other times it is hidden. In particular, this process occurs when a COBOL program does an OPEN OUTPUT with ACCESS IS DYNAMIC or ACCESS IS RANDOM specified. What COBOL does in this circumstance is load a record with a key of binary zeros and then close the file. COBOL subsequently reopens the file and deletes the record. This effectively places the file in a loaded but empty state.

While this situation is handled better by IAM Enhanced format files with variable overflow, than it was handled with Compatible format files, the basic problem still exists in that the records that are mass inserted into the file are contained within the overflow areas, Extended Overflow and Extended PE. The storage required for the index to these overflow areas can be substantially more than if the file had been fully loaded rather than single record loaded. If the single record load process cannot be changed, then it is recommended that the dataset be reorganized after the mass inserts are done. This will reduce overall virtual storage requirements, and provide for more efficient processing.

JCL In general, there are no required JCL changes from file loads that were using VSAM clusters. The DD card for the IAM dataset should specify only the DSN= (Dataset name) and a DISP=OLD (disposition). A DISP=SHR is allowed as well, although DISP=OLD is recommended because only the program loading the dataset is allowed to have the dataset open while the load is in progress. IAM will enforce this restriction using an ENQ mechanism with the major name of IAMENQ. The AMP parameters may be specified, however there definitely must not be any DCB parameters. The VOL=SER and UNIT parameters should not be specified either.

When the IAM dataset is defined and loaded in the same IDCAMS step, the REPRO should use the OUTDATASET (ODS) parameter, not the OUTFILE parameter to reference the IAM dataset. This is because if the IAM dataset was defined to a different volume than the one it existed on when the job step was started, then the DD card is no longer valid because it has the old volume(s) allocated to it. This guideline has to particularly be followed by installations that are using DFSMS managed datasets, or some other allocation products such as BMC's MAINVIEW (formerly POOLDASD), CA-ALLOCATE, or ACC/SRS.

The IAMINFO DD is optional, but highly recommended. A report will be produced to that DD when the IAM file is closed, containing the dataset attributes and statistics about the load. The IAMOVRID DD statement is only necessary when providing IAM overrides for this job step. The IAM Overrides that are applicable to the file load are described below.

10.30 CONTINUED . . .

OVERRIDES FOR DATASET LOAD

There should only be an infrequent need to override the file load process. The main reasons for using an override for the load process are to use the IAM feature of reorganizing the file without decompressing the data, to alter the buffering for the file load, to increase the size of the data space used to temporarily hold the index, or to specify certain attributes if the dataset had been defined through JCL. The BACKUPCOMPRESSED, CRBUFOPT, and the DATASPACE keywords are only applicable to the file load process, and when needed must be specified for the load.

The following list contains the CREATE Override keywords which are applicable to the file load process. For the most part, the CREATE Override keywords have the same meaning and implication when used during the file load, as they do during the file define. The underscored portion of the keywords indicates their minimum abbreviation. The keywords applicable to the define process are:

Keyword	Brief Description
BACKUPCOMPRESSED	Specifies that the input data is already in an IAM Data Compressed format. This can only be specified on a file load step with data that was created with the IAM ACCESS Override of BACKUPCOMPRESSED.
BLKSIZE=nn	Specifies the block factor (1 - 15) or block size of the IAM dataset.
COMPATIBLE	Specifies that IAM is to create a compatible format dataset. A compatible file format has overflow areas that are formatted when the file is loaded, and are fixed in size until the file is redefined or reorganized.
CRBUFOPT=	Specifies the buffering option to be used for the file load. **CYL –** Buffers for one cylinder's worth of blocks is acquired, approximately 1/2 cylinder is written per I/O. **MCYL –** Buffers for two cylinder's worth of blocks is acquired, one cylinder is written per I/O. **MTRK –** Buffers for ten tracks are acquired, five are written per I/O. **TRK –** Buffers for two tracks are acquired, one track is written per I/O. Default value is MCYL
DATACOMPRESS= [YES|NO|HW]	Indicates whether IAM is to data compress this dataset. Innovation recommends that data compression be used for datasets that exceed 5 cylinders of DASD space, which is the default in the IAM Global Options Table. Hardware compression can also be selected here, if not done so during the file definition.
DATASPACE=	Specifies the size, in megabytes, of the Data Space to be used for the temporary storage of the index to the IAM file that is being loaded. Valid values are from 0 to 2048. A value of 0 results in the use of a dynamically allocated temporary dataset. The default value is 256.
DDNAME=	Specifies which dataset the override is applied to. This will indicate the name of the DD statement being opened for the file load. If this value is set to &ALLDD, then the overrides are applicable to any IAM file that is not otherwise explicitly overridden.
DICTIONARY=	Specifies the four character suffix for the name of the user provided hardware compression dictionary. The dictionary must be in load module format, with the first four characters being 'IAMD'. Review the section on Hardware Compression for information on creating and naming the compression dictionary. The default is to use the IAM provided dictionary.
DSN=	As an alternative to DDNAME=, specifies the dataset name that the override data is to be applied to. Either DSN= or DDNAME= must be specified.

10.30 CONTINUED . . .

ENHANCED	Specifies that IAM is to create an Enhanced format IAM dataset. This type of dataset uses a dynamic overflow area, which can acquire additional DASD extents during file updates, as needed. This is the default value, unless otherwise specified or changed in the IAM Global Options Table.
INTEGRATED=nn	Specifies CI% free space, especially useful for data compressed ESDS datasets.
MAXBUFNO=nnnn	Specifies the default maximum number of buffers to be used during file access for this dataset.
MAXSECONDARY=nn	Specifies a default value from 0 - 10 as a multiplication factor for the secondary space quantity, when the dataset exceeds five extents on a volume. When overridden on the load, this will be the value for extents taken only during the file load.
MINBUFNO=nnn	Specifies the default minimum number of buffers to be used during file access for this dataset.
MULTIVOLUME= **[PRIMARY | SECONDARY]**	Specifies space quantity to request when IAM believes that the next extent will be placed on the next candidate volume.
PSEUDORBA	For ESDS files, indicates that the file can exceed 4 gigabytes of user data. IAM generates RBA values that are different than normal VSAM values. Can not be used with SAP, or other software that depends on the normal VSAM values.
RELEASE=[YES|NO]	Indicates whether or not IAM is to release unused and unreserved DASD space whenever the file is loaded. Default is to release space on the first load only, and only when a secondary space value is specified. COMPAKTOR will also release unused space for IAM enhanced format datasets until the file has expanded into the extended overflow areas of the file.
VAROVERFLOW=[YES|NO]	For Enhanced format files, indicates that IAM can use true variable length records in the Extended Overflow area. This will provide for more effective use of DASD space. The default value is YES.
XESDS	For ESDS files, indicate that IAM is to use an 8-byte RBA value. This is consistent with the VSAM Extended Addressability support. Default is that IAM will use a 4-byte RBA. The 8-byte RBA is a way for ESDS files to exceed 4 gigabytes of data, but application programs must be able to handle the 8-byte RBA.

10.30 CONTINUED . . .

EXAMPLES OF LOADING AN IAM DATASET

Below are a couple of examples using IDCAMS to load an IAM dataset. As stated above, IAM datasets can be loaded by any program that can load a VSAM cluster. For conversions from VSAM clusters to IAM datasets, there rarely will be any need to change the JCL, except perhaps to add an IAMINFO DD for SYSOUT to get the detailed report of the file load.

EXAMPLE A: LOADING AN IAM FILE WITH AN OVERRIDE

In the example below, an IAM dataset is loaded with IDCAMS. An IAMINFO DD statement has been added to obtain the run time file load report from IAM. The OUTFILE parameter is used to specify the output IAM file, which also corresponds to the ddname specified on the CREATE Override statement provided. The override is used to specify the maximum buffering for the file load, so that it will run as fast as possible. Also note the REUSE is specified on the REPRO command, which will cause the dataset to always be reloaded, even if it already contains data.

```
//LOADFILE EXEC PGM=IDCAMS
//SYSPRINT  DD  SYSOUT=*
//IAMINFO   DD  SYSOUT=*
//INFILE    DD  DSN=my.seqfile,DISP=SHR
//IAMFILE   DD  DSN=my.iamfile,DISP=OLD
//IAMOVRID  DD  *
  CREATE    DD=IAMFILE,CRBUFOPT=MCYL
/*
//SYSIN     DD  *
  REPRO INFILE(INFILE) OUTFILE(IAMFILE) REUSE
/*
```

Figure 23: Example of Loading an IAM Dataset (EX1030A)

EXAMPLE B: LOADING COMPRESSED DATA INTO AN IAM DATASET

In this example, the OUTDATASET (ODS) is used instead of OUTFILE on the IDCAMS REPRO. This will cause IDCAMS to dynamically allocate the IAM dataset. Because the ddname is not known in advance, the CREATE override specifies the dataset name (DSN=). In this example, the sequential input dataset was previously created by copying an IAM compressed dataset with the BACKUPCOMPRESSED override, so it must be reloaded with that same override specified on the CREATE override. A large BUFNO value is specified on the DCB for the input file to provide for faster processing, along with the CRBUFOPT override.

```
//LOADCOMP EXEC PGM=IDCAMS
//SYSPRINT  DD  SYSOUT=*
//IAMINFO   DD  SYSOUT=*
//SEQFILE   DD  DSN=my.compresd.seqfile,DISP=SHR,DCB=BUFNO=60
//IAMOVRID  DD  *
  CREATE    DSN=my.iamfile,BACKUPCOMPRESSED,CRBUFOPT=MCYL
/*
//SYSIN     DD  *
  REPRO INFILE(SEQFILE) ODS(my.iamfile) REUSE
/*
```

Figure 24: Example of Loading an IAM dataset with compressed data (EX1030B)

10.40 PROCESSING IAM DATASETS

OVERVIEW

After the IAM dataset has been defined and subsequently loaded with data, the IAM dataset is ready to be used for typical online and batch processing. Processing an IAM dataset instead of a VSAM cluster will generally significantly reduce CPU times, physical I/O's, and elapsed times for most programs. The IAM VSAM interface was developed with the intent of eliminating, as much as possible, the need to change any program, JCL, or CICS tables. On rare occasion, there may be a need for a change, which can frequently be accomplished with an IAM override. IAM supports almost all of the typical VSAM processing options, including specification of LSR pools, user exits specified via the VSAM EXLST macro, asynchronous I/O, and so forth.

IAM does not support RBA or control interval access of KSDS types of files, and does not support control interval updates for ESDS type files.

OPEN PROCESSING

As part of the functioning of IAM's VSAM Interface, IAM will screen all open requests for VSAM ACB's. If the ACB being opened is for an IAM dataset, IAM routes control to the appropriate IAM module to perform the OPEN processing. If the open is not for an IAM dataset, the IAM routes control to the IBM VSAM open processing modules, and does not further screen or interfere with VSAM open or I/O processing.

One of IAM's major features is that the index to the file is kept in virtual storage while the file is open. This feature not only eliminates I/O for the index component, but also provides for savings in CPU time. IAM can always get to any specifically requested record with no more than one I/O. Because the index is read into storage at open time, open processing for an IAM dataset may take longer than a VSAM open. An IAM open also involves acquiring virtual storage required for processing, acquiring buffers, and setting up the I/O control blocks, and loading the necessary IAM load modules. IAM's self-defining file structure eliminates the need for retrieving information about the dataset from the system catalog, which eliminates that VSAM overhead.

Reading the index to the prime area and the prime extension is a straightforward process. Building the overflow index involves more processing, because the overflow index itself is never saved within the IAM dataset. The overflow index contains one entry for each record in overflow, unlike the prime and prime extension index that contain the high key in each block. To build the overflow index, IAM must read all of the used overflow blocks, determine the key of each overflow record, and insert the key into the proper position in the overflow index. As more and more records are put into overflow, this process will take longer to complete. Regular file reorganizations will help to keep the open processing time at a minimum.

IAM Open processing also automatically performs an implicit verify operation. On files that have been opened for update, IAM will update the extended index and end of file information if necessary.

Once the file open process has completed, IAM is ready to handle the application I/O requests.

I/O REQUEST PROCESSING

I/O to an IAM dataset is performed by using the standard VSAM application programming interface (API). Standard VSAM protocol is used by IAM, including error reporting. IAM does not abend when a I/O request fails, the appropriate return codes and failure codes are provided, and the appropriate error exit routine is also invoked if so specified by the calling program. As with VSAM, it is the responsibility of the application program to determine the appropriate action for any particular type of error. The various error codes set by IAM are documented in section 80.21 of the IAM manual.

For KSDS types of files, IAM handles all requests except those that specify Relative Byte Addressing (RBA) or Control Interval access. Such requests are failed with an error code of 104 (x'68') and a return code of 8. For ESDS files, IAM does not support Control Interval updates, which will fail with an error code of 68 (x'44') and a return code of 8.

10.40 CONTINUED . . .

BUFFERING IAM's dynamic buffer management technique, Real Time Tuning, is automatically working in response to the application I/O requests. The IAM buffering technique dynamically adjusts buffering and I/O techniques, along with adjusting the number of buffers, based on the types of requests being issued by the application. IAM takes the guesswork out of the tuning process. No need to be concerned about whether or not to use LSR type of buffering, or how many buffers are needed for optimum I/O performance. IAM will select the buffering technique and quantity, within specified or defaulted limits, automatically. Buffering and I/O techniques are adjusted dynamically in response to the application requests.

Real Time Tuning works within a range for quantity of buffers, referred to as MINBUFNO and MAXBUFNO. These ranges can be specified through the IAM Override facility, using the MINBUFNO and MAXBUFNO keywords. Default values for MINBUFNO and MAXBUFNO are determined from the IAM Global Options Table. Using the default Global Option values will eliminate the need for increasing the MAXBUFNO value for all but the most heavily accessed datasets. The MAXBUFNO value can also be increased by the BUFND parameter specified in JCL as a subparameter of the AMP parameter, or within the program in the ACB.

For more information on IAM's Real Time Tuning capability, refer to section 02.20 of the IAM Manual.

DYNAMIC TABLING IAM has a special feature that keeps the most frequently referenced randomly read records in virtual storage. This facility provides substantial benefit to some applications, by eliminating physical I/O as records are repeatedly retrieved from the table. Applications that may potentially benefit from this feature are those that have a high volume of GET RANDOM requests, combined with a very low quantity of PUT UPDATE and ERASE requests. The update requests are detrimental because they will force IAM to read the data block into storage, to perform the update. The other request types will not benefit from nor hinder the use of the dynamic table.

This capability is easily used. Simply specify the table size, in kilobytes, using the IAM DYNCORE override. Statistics on the DYNCORE usage are provided in the IAMINFO report.

FILE EXPANSION IAM Enhanced Format files are dynamically expandable, through the use of secondary extents. IAM files that are not DFSMS Extended Format can be extended up to a maximum of 16 extents per volume, and a maximum of 255 extents across multiple volumes. IAM files that are DFSMS Extended Format can be extended up to a maximum of 123 extents per volume, for up to 59 volumes. Extended Format datasets do not have an explicit total maximum number of extents limitation. As records are added, or updated records increase in length, additional DASD space is acquired and formatted as needed. This expansion area is called the IAM Extended Area. The Extended area consists of Extended Overflow blocks and Extended PE blocks. The Extended PE blocks are used to handle the addition of records with key values higher than the highest key value in the dataset, while the Extended Overflow blocks will handle all of the other file expansion needs.

For IAM files that are not DFSMS Extended Format, if they experience larger than anticipated growth, IAM can dynamically increase the size of the secondary space being requested. Once a dataset has five extents on any particular volume, IAM will increase the secondary space by the MAXSECONDARY factor, providing that the value does not exceed the original primary space requested, and that there is sufficient space on the volume for the desired quantity.

INDEX SPACE IAM has the optional capability to place the bulk of the index for IAM datasets in a data space. This can result in a significant reduction in virtual storage requirements within the address spaces using IAM datasets, and is especially helpful for large online systems. This feature defaults to being active for CICS regions, which is controllable through the IAM Global Options Table and IAM ACCESS Overrides.

10.40 CONTINUED . . .

CLOSE PROCESSING

Once processing has been completed for the IAM dataset, the application program should explicitly close the dataset. If the dataset is not explicitly closed, then task termination will close the dataset when the task that opened the dataset completes. The basic functions of close processing are to:

- Freemain storage used to access the dataset.
- Delete the processing modules used for accessing the dataset.
- On files opened for update, close processing will write out any updated blocks to the dataset that have not yet been written, and update the file statistics and control information.
- Close the internal DCB used by IAM to access the dataset.
- If an IAMINFO DD is present, format and produce the IAMINFO report.
- Produce an SMF record of file activity, if so enabled by the IAM Global Options.
- Disconnect the ACB from the control blocks built, and from the system.

Once the dataset is closed, no further I/O activity can take place. The application program should check the return code from the close to make sure that it was successful. A return code of 4 indicates that the close failed. It is rare that a close will fail, but it can happen. The main reason for a close to fail is that there is insufficient storage for the close to process. This is indicated by an error code of 136 (x'88'), that is returned in the ACB error code field. If this error occurs, the application should freemain some storage, or close other files, then reattempt to close the IAM file.

FILE FULL OR SX37 ERROR

On IAM files opened for update, a file full error is not considered by IAM to be a fatal error. The Sx37 abends will be masked out by IAM for Enhanced Format IAM datasets, and treated as a file full logical error. The updated or inserted record will not be written out to the dataset. For updated records, the non-updated version of the record will remain in the dataset. The error conditions will be reported back to the calling program using standard VSAM logical error codes, and exits where applicable. To prevent getting into a reoccurring Sx37 loop, IAM will not try to take any additional extents for a file once such an error occurs, unless the dataset is closed and reopened.

File full errors can also occur when IAM is unable to obtain additional virtual storage for the expanding index structure. This type of file full condition is accompanied by an IAMW03 error message.

JCL SPECIAL CONSIDERATIONS

In general, there are no JCL changes required to access IAM files in place of VSAM. The only required JCL parameters on the DD card for an IAM file are the DSN and DISP. If the JCL was set up to use Batch LSR for VSAM, that can be left in place for IAM, although IAM will not use any of the VSAM LSR buffers. Likewise, any AMP parameters can also be left in place. The BUFNI value will always be ignored by IAM, because it does not need or use index buffers. The BUFND value will be looked at and used as the value for MAXBUFNO, providing that there was no MAXBUFNO override, and that the value exceeds the default value for MAXBUFNO. STRNO will be used to establish the initial number of place holders, as done by VSAM. With IAM Enhanced format datasets, IAM now has dynamic string acquisition so STRNO is not as critical as it was for Compatible Format IAM datasets.

While optional, users are highly encouraged to add an IAMINFO DD statement allocated to SYSOUT to job steps that are using IAM datasets. When this DD statement is present, a one-page report will be produced each time an IAM dataset is closed, providing file characteristics and statistics from the execution, including logical and physical I/O counts, and storage requirements.

10.40 CONTINUED . . .

MULTIPLE ACB IAM datasets can be processed by multiple ACB's opened within the same address space, either with the same or different DD names. IAM will share the control block structure and the index between the different ACB's, as long as all of the ACB's are opened under the same task.

The advantages of sharing the control block structure and index are that doing so significantly reduces the storage required to process the dataset with the multiple ACB's, plus it provides for complete data integrity between the different ACB's. If the sharing is not done, then each ACB is completely independent, and will be subject to the standard Cross Region sharing considerations. This means that unless a file is defined with Share Option 3 or 4, only one of the ACB's can be opened for update.

Another advantage of the sharing is that for CICS, or other long running jobs, there is provision for 24-hour operation, if IAM RLS is not being used. This is achieved by having one of the ACB's opened for read only access, and the other being opened for update. When the update ACB is closed, all of the updated data buffers are written out, and the necessary control information in the dataset is updated. The dataset can then be updated by other jobs. Then, when the update ACB is reopened, all the buffers are invalidated to insure that only the current blocks from the dataset are used, plus the overflow and PE indexes are rebuilt to reflect the current state of the file. This will provide for online read only access to the file while it is being updated by batch processing.

RLS: RECORD LEVEL SHARING Enhanced format IAM files may also be processed using IAM RLS to provide for data integrity with concurrent update jobs executing under the same MVS image. It is expected that such processing will be handled by automatic dataset selection criteria that an installation provides to IAM. The use of IAM RLS is intended to be completely transparent to the jobs or CICS regions that may have files eligible for IAM RLS processing. Customers can use the SHAREOPTION override to control RLS processing, if the automatic selection includes the use of SHAREOPTION as part of the criteria. The default is that files defined or accessed with share options of either (3,n), or (4,n) are automatically eligible for IAM RLS processing. Alternatively, customers may use the RLS or NORLS Override keywords to direct RLS processing. Full information on the use of IAM RLS (Record Level Sharing) is in section 20 of this manual.

JOURNALING IAM can also provide a journal for file update activity to facilitate recovery with the IAMJREST program. Journaling is activated based on either the JRNAD= override, or for DFSMS managed datasets by the LOG parameter on a DEFINE CLUSTER. Journaling can be performed either as a normal access function, or through IAM RLS. The normal access will have a separate journal for each IAM dataset, whereas the journals under IAM RLS contained the combined records from all datasets that are being journaled. Further information on journaling and recovery is in section 10.88 of the IAM manual for normal access, or in section 20 for IAM RLS access.

10.40 CONTINUED . . .

OVERRIDES FOR IAM FILE PROCESSING

IAM ACCESS OVERRIDES IAM has an override facility that provides easy access to many of the IAM special features. Some of the more common reasons for using IAM Overrides include:

- Adjusting the range of buffers for IAM Real Time Tuning
- Enabling use of the IAM Dynamic Tabling feature
- Enabling use of IAM's Index Space
- Enable or disable the IAM journaling
- Altering values for IAM's Dynamic Region Adjustment
- Activating the BACKUPCOMPRESSED feature
- Altering IAM's Dynamic Secondary Space adjustment feature values

HOW TO USE OVERRIDES Use of the IAM Override facility is easy. Just add an IAMOVRID DD card to the job steps that are processing IAM datasets, and provide the simple control card image input with the desired parameters. The IAMOVRID DD is normally a DD * that is followed by the control cards within the input job stream. The IAMOVRID could also be a sequential or partitioned dataset member that contains the input control cards. The format of the override card for the ACCESS overrides are that they begin with the word ACCESS, which can begin in column one or be preceded by blanks, is followed by one or more blanks, then the override keywords with their values are specified, separated by commas. The keywords and their values may not go beyond column 71. Continuations, if needed, are indicated by leaving a comma after the last keyword on the preceding card, which is followed by other card(s) with the additional keywords. Full details on the IAM Overrides can be found in Section 30 of the IAM User's Manual. Shown below are the commonly used IAM ACCESS Overrides that are applicable to IAM's Enhanced Format Files.

10.40 CONTINUED . . .

OVERRIDE KEYWORDS

Keyword	Brief Description
BACKUPCOMPRESSED	Indicates that IAM is not to decompress the data when passing the requested records to the requester. The data will remain in an IAM data compressed format, which will not be usable by application programs for purposes other than copying and reloading an IAM data compressed dataset. This option can not be used with Syncsort.
DDNAME=	Keyword that specifies the DD name(s) of the files to which the override values are to be applied. The special value &ALLDD can be used to apply the overrides to all files except those that are otherwise explicitly specified. Either DDNAME or DSN is required.
DSN=	Specifies the name of the dataset to which the override values are to be applied. This can be used for Enhanced Format files only. Either DDNAME or DSN is required.
DYNCORE=nnnn	Specifies the amount of virtual storage, in kilobytes, to be used for IAM's Dynamic Tabling facility. Maximum value is 16000.
INDEXSPACE=[YES | NO]	Indicates whether or not IAM is to use a data space for the prime and overflow index structures, rather than extended private.
JRNAD=	For Enhanced format files, specifies the IAM journaling capabilities to be used. When specified on the ACCESS override statement, if the value specified enables journaling, the value will be combined with any specification from the CREATE override when the file was defined or loaded. For files not being processed by IAM RLS, users must pre-allocate and catalog a log dataset to be used for the journaling, which is required to be the name of the IAM dataset / cluster, appended with the characters ".LOG". IAM journaling will not be active during file loads, reorganizations, or during recovery from the journal. For files being processed by IAM RLS, the users must specify the journal datasets to the IAM RLS startup procedure if they want to use journaling, and had not specified to do so when the dataset was defined or loaded. The ACCESS override only changes the journaling value for the job step on which it is specified. Valid values are: **BOTH –** The IAM log dataset will contain both before and after images. This will enable the user to perform either a forward recovery, or a backward (backout) type of recovery. **BEFORE –** The IAM log dataset will contain before images of updated records. This option allows backward (backout) recoveries only. If AFTER images were specified on a CREATE override, then AFTER images will still be logged to the journal. **AFTER –** The IAM log dataset will contain after images of updated records. This option allows forward recoveries only. If BEFORE images were specified on a CREATE override they will still be logged to the journal. **NONE –** The IAM journaling feature will not be used for this IAM dataset. This will turn off any journaling for this job step, even if journaling had been specified on a CREATE override. Default value is NONE, unless otherwise specified when the file was defined or loaded.
MAXBUFNO=nnnn	Specifies the maximum number of buffers IAM will use for the specified file. Maximum value allowed is 2048. (Note: maximum for compatible format files is 32.)

10.40 CONTINUED . . .

MAXREGION=nnnn Specifies the maximum value, in megabytes, which IAM Dynamic Region Adjustment will allow the Extended Private region to be set to. Maximum value allowed is 2048.

MAXSECONDARY=nn This applies only to datasets that are not DFSMS extended format. Specifies the multiplication factor to be used on the secondary quantity when a dataset exceeds 5 extents on a volume. Maximum value that can be specified is 10. Note that IAM will never increase the secondary space to a value greater than the primary space.

MINBUFNO=nnnn Specifies the minimum number of buffers that IAM is to use for the specified dataset. If specified, this will also be the number of buffers that IAM initially acquires during open.

NORLS A keyword that specifies that IAM is not to process this file under the IAM RLS address space, even if the installation's RLS selection criteria indicates that the file(s) are eligible for RLS processing.

REREAD Indicates that IAM will always reread the overrides each time an IAM file is opened. Normally, IAM will only read the overrides on the first open, and save all the values in virtual storage for subsequent reference. Specifying this keyword allows the overrides to be changed for long running online systems.

RLS A keyword, that when specifies, indicates that the file(s) are to be processed by the IAM RLS address space. If the IAM RLS address space is not active, the OPEN of the file will fail.

SHAREOPTION=n Overrides the VSAM Cross Region Share Option to use for this execution. Provided as the alternative to performing an ALTER on the file to change the Shareoptions.

10.40 CONTINUED . . .

EXAMPLES OF ACCESSING AN IAM DATASET

A few JCL examples of processing an IAM dataset are shown below. Essentially, there are no JCL changes to use an IAM dataset instead of a VSAM dataset, except when one desires to obtain the IAMINFO report, or wants to provide some IAM Overrides.

EXAMPLE A: BASIC ACCESS OF AN IAM FILE

This example demonstrates the simplicity of using IAM. The only change made to this job stream when converting to VSAM was the optional addition of an IAMINFO DD card, to obtain the IAM run time reports. Note that the VSAM AMP parameter has been specified for one of the IAM datasets. The BUFNI specification is ignored by IAM, however the BUFND value may be used, if it is higher than the default for MAXBUFNO.

```
//PROCESS  EXEC PGM=anypgm
//SYSPRINT DD   SYSOUT=*
//FILEA    DD   DSN=prod.iam.filea,DISP=SHR
//FILEB    DD   DSN=prod.iam.fileb,DISP=SHR
//FILEC    DD   DSN=prod.vsam.filec,DISP=SHR
//MASTER   DD   DSN=prod.iam.master.file,DISP=SHR,
//              AMP=('BUFNI=6','BUFND=60')
//IAMINFO  DD   SYSOUT=*
```

Figure 25: Example of Processing an IAM Dataset (EX1040A)

EXAMPLE B: USING IAM ACCESS OVERRIDES

This example demonstrates how to use the IAM Override facility when processing multiple IAM datasets. Because the job is processing several datasets, some of them with large indexes, the INDEXSPACE=YES override is being specified with DD=&ALLDD, so that the virtual storage for the index will not come out of the job's address space.

The TABLE file is a read only reference KSDS dataset that is used for data validation. The records in this dataset tend to be on the small side, less than 100 bytes, and processing is random against a subset of the records, which tend to be referenced many times. For this dataset, the IAM Dynamic Table Facility is quite useful. An IAM Override is being used to enable IAM's Dynamic Tabling Facility, with a table size of 128K. The INDESSPACE override is also being specified for this dataset, because it is being explicitly overridden and the value will not be picked up from the DD=&ALLDD override card.

```
//MASTRUPD EXEC PGM=updpgm
//SYSPRINT  DD  SYSOUT=*
//MASTER    DD  DISP=SHR,DSN=prod.master.iam.file
//SEQFILE   DD  DISP=OLD,DSN=prod.seqfile.iam.esds
//TABLE     DD  DISP=SHR,DSN=prod.table.iam.file
//FILEA     DD  DSN=prod.iam.filea,DISP=SHR
//FILEB     DD  DSN=prod.iam.fileb,DISP=SHR
//FILEC     DD  DSN=prod.iam.filec,DISP=SHR
//IAMOVRID  DD  *
  ACCESS    DD=TABLE,DYNCORE=128,INDEXSPACE=YES
  ACCESS    DD=&ALLDD,INDEXSPACE=YES
/*
//IAMINFO   DD  SYSOUT=*
```

Figure 26: Example of Processing IAM Datasets with Overrides (EX1040B)

10.50 IAM TUNING GUIDELINES

OVERVIEW For the great majority of datasets converted to IAM, there is no real need to perform any tuning. This is because IAM, with its Real Time Tuning capability, can generally provide an outstanding level of performance, without the need for any manual intervention. The tuning guide is being provided for those installations that want to make sure that they are getting the best possible performance from IAM. It is also being written to address those few files that just seem to require a bit of extra effort. The last reason is to provide an aid for installations that are having a resource constraint, particularly with real or virtual storage.

What do we mean by performance? With an access method, such as IAM, performance is retrieving and storing data faster while using less computing system resources. Nothing is free however, and frequently there are tradeoffs involved, such as using more of one resource to use less of another. These resources include the utilization of the processor, referred to as CPU time, DASD space utilization, utilization of channels, control units, and physical devices to move the data between processor storage and the device, and the use of both virtual and real storage. An access method needs to use a portion of all of these resources to provide the service. Frequently tuning involves adjusting resource use from a constrained resource to an unconstrained resource.

The longest portion of any I/O operation is the time it takes to transfer data from the storage device into processor storage. It therefore follows that the fastest logical I/O is one where there is no physical I/O. IAM's goal is to satisfy as many logical I/O requests as possible without performing any physical I/O. IAM generally utilizes virtual storage, and of course the underlying real storage, to obtain the high performance. The two primary storage areas to accomplish that are the index area and the buffers. IAM retains the entire index, normally in an internally compressed format, for the dataset in virtual storage while the dataset is open. This eliminates any need for physical I/O to read the index. Any record can be retrieved with no more than 1 physical I/O. With VSAM, assuming the required control intervals are not within the buffer pool, a typical random read will generally require at least 2 physical I/O's, one for the index and the other for the data. Depending on the VSAM index structure, more I/O's may be required for the index component.

The other way IAM reduces physical I/O's is through it's proven Real Time Tuning buffer management algorithms. Using dynamic buffer management techniques, IAM is able to reduce physical I/O to a level that is quite difficult to achieve with VSAM, even with extensive manual tuning. IAM will dynamically adjust the number of buffers, the buffer management algorithms, and physical I/O techniques used to match the application programs current requirements. The result with IAM is better performance, resulting in reduced elapsed times for batch jobs, reduced response times for online transactions, and less time investment required on tuning to achieve those goals.

10.50 CONTINUED . . .

RECOMMENDED GLOBAL OPTION SETTINGS

The easiest way to minimize the need for manual tuning is to use a good base of default parameters. With IAM, these parameters are provided through the IAM Global Options Table. The default options will provide a high level of performance for the great majority of IAM datasets. Even better performance may be easily achievable by making some changes to the default options. The recommended values below will generally improve performance for many installations, providing that they are not constrained with various resources, as identified with each recommendation.

DATA COMPRESSION IAM defaults to using data compression for any dataset that is defined as being 75 tracks (5 cylinders on a 3380 / 3390 device) or larger. This is one of the best IAM facilities and we strongly recommend that data compression be left enabled. The benefits are reduced DASD space requirements, reduced virtual storage requirements for indexing the dataset, and reductions in physical I/O because there is effectively more data in each block. The cost is additional CPU time to process the dataset. For many datasets, IAM CPU time with data compression is still lower than VSAM without compression.

IAM Version 8.0 can also use the IBM hardware compression instruction, which on the zSeries processors will have CPU times for IAM files that are similar to IAM software compression. For some datasets, the hardware compression may achieve more space savings than with the IAM software compression, particularly if a customized data dictionary is built. If a dataset is quite large, and heavily accessed, then consideration should be given to using hardware compression.

Unless CPU processing time is very constrained or your processing cost is based on CPU time and EXCP counts are not, Data Compression should be left enabled.

BUFFER SPACE In Version 8.0, IAM defaults to **BUFSP=896000**. IAM calculates the default MAXBUFNO from this Global Options Value. This recommended value causes MAXBUFNO to go up to just over one cylinders worth of buffers on both 3380 and 3390 types of devices. This will be of particular benefit to batch jobs that read IAM datasets sequentially, in that they will be able to read in a full cylinder worth of blocks per I/O, plus have some extra buffers to handle Extended Overflow blocks. This may also provide benefits to other jobs that process the files randomly do to the larger MAXBUFNO default value.

For CICS regions, there is the CICSBUFSP parameter, which defaults to 262144 (256K). This value will be used by IAM to calculate MAXBUFNO when running under CICS. This Global Option was added so that customers could give their batch work a larger quantity of buffers automatically, while not giving as many to CICS due to the frequent virtual storage constraints that are encountered under CICS.

FILE LOAD BUFFERING The next recommendation is to leave CRBUFOPT=MCYL, to cause a file load to buffer up to one cylinder worth of data, while physically writing a full cylinder of data. This is the optimal buffer setting for file loads.

ESDS PROCESSING If your installation is using IAM ESDS types of datasets that are updated, then be sure to set ESDSINTEGRATED=5 or higher. This will allow some room for record updates that require more space after data compression without having to use the Extended Overflow area. The cost is more DASD space usage to load the dataset initially, but if the dataset is updated, this will prevent the use of DASD space, virtual storage, and I/O for the Extended Overflow area.

VARIABLE OVERFLOW The other change is to leave VAROVERFLOW=YES. This enables IAM's variable overflow that will result in more data records in each overflow block. This should reduce DASD space requirements for the overflow area, plus may also help reduce physical I/O's to the overflow area. The savings, particularly for datasets defined with very large maximum record lengths can be substantial. The disadvantage is that records that are repeatedly updated may have to be moved to a different overflow block from time to time if the record length increases. If you are migrating from Version 6.3, then this option should not be set until you have IAM Version 8.0 in production on all of the systems using IAM.

10.50 CONTINUED . . .

GENERAL CONSIDERATIONS FOR OPTIMUM PERFORMANCE

Most of the IAM datasets will achieve outstanding performance results, particularly with the recommended Global Options settings. If you want to make sure that you are receiving the best performance possible with all of your datasets, or are having some type of performance problem, then consider the following general guidelines.

1. Make sure that you have a way to obtain IAMINFO reports. This can be done either by adding an IAMINFO DD card to the JCL for job steps using IAM files or by collecting the IAM SMF records, then post processing the data with the IAMINFO command of the IAMSMF utility program. These reports contain critical information for detailed tuning. Become familiar with the contents of these reports, as they provide a lot of useful information. By activating and collecting the IAM SMF records, you can also utilize the IAMSMFVS reports for a more concise report format that will make it easy to find those datasets that might require additional attention.
2. Periodically review the IAMINFO reports. If more buffers would have helped reduce physical I/O's, the IAMINFO report will contain an IAM368 message indicating so. In fact, you can request that IAMSMF print off only those IAMINFO reports where that message appears, with the keyword ATTRIBUTE=MOREBUFFER. If this message is appearing for several datasets, then perhaps the BUFSP Global Option should be increased to avoid the need for several overrides.
3. Use data compression. This will help reduce physical I/O's, reduce virtual storage for the prime index, and reduce DASD space requirements.
4. Make sure that heavily updated files are regularly reorganized. This will help prevent virtual storage problems, long open times, and high physical I/O activity.
5. For datasets with large maximum record sizes, be sure to activate the Variable Overflow feature. This feature can result in significant savings of DASD space and I/O by increasing the number of records that are stored within each overflow block.
6. Avoid the specification of Share Options 3 or 4 for IAM datasets, unless you have activated IAM RLS. IAM does not support sharing files for update across multiple systems only within a single system with IAM RLS active. Specification of those share options without IAM RLS will force additional physical I/O's that can be substantial.
7. Investigate increasing the block size for datasets with a large Prime Index structure, particularly if the dataset has relatively large record sizes.

10.50 CONTINUED . . .

BUFFERING IAM makes it easy to determine when more buffers could have reduced I/O by providing the IAM368 message in the IAMINFO report. Unless there is a concern about storage, there is no reason to be concerned about being overly aggressive at setting MAXBUFNO. IAM's Real Time Tuning will carefully adjust the buffering for the dataset as processing needs vary. For programs that do all sequential processing the maximum number of buffers used for the file will be the number of blocks per cylinder plus a few additional buffers to handle Extended Overflow blocks. Usually providing one or two tracks worth of buffers for overflow will be sufficient, unless a dataset has a very extensive use of overflow. Using the default BUFSP Global Option setting will handle setting the defaults to maximize the buffering for sequential processing, eliminating the need to increase buffers for most batch jobs.

For programs that do all random I/O, a mix of random and sequential I/O, or short strings of sequential I/O requests, then the MAXBUFNO value should be increased by a quantity that you feel comfortable with. If you are not concerned about virtual storage usage or paging, then use a large quantity. If however storage is of a concern, increase the value by 4 or 8, and see how that helps. The methods of increasing MAXBUFNO for any particular file include:

- Providing an IAM ACCESS MAXBUFNO override for the job step and dataset.
- Specify the BUFND parameter, either within the ACB or as part of the AMP parameter on the DD card for the dataset, e.g. AMP=('BUFND=nnn'). For CICS files not in a LSR pool, the resource definition for the data buffers will result in changing the BUFND value in the ACB that is used by CICS.
- To raise buffering for all jobs that use this dataset, provide an IAM CREATE MAXBUFNO override when the dataset is defined, loaded, or reorganized. The specified MAXBUFNO value will be applied whenever the dataset is accessed.
- Specify a value for BUFSPACE on the IDCAMS define control statement for the dataset.

The two circumstances where you might not want to increase buffers for the job are:

1. When the job is performing sequential processing against a dataset that is concurrently open to online systems. This is because the batch job could end up dominating the file, resulting in poor response times for users of the dataset on the online system. In fact, you will probably want to reduce MAXBUFNO for such jobs.
2. When the job has a virtual storage constraint. Refer to the section on Storage Usage for adjusting buffers with jobs that have storage constraints.

Otherwise, it is perfectly fine to increase the MAXBUFNO value.

10.50 CONTINUED . . .

EXTENDED OVERFLOW

Excessively large Extended Overflow usage can result in a deterioration of performance that can usually be avoided. These problems are avoided by the periodic reorganization of files when they are using some large quantity of extended overflow. VSAM datasets also require reorganizations due to performance deterioration and space usage. Because of that, many application job streams that were converted to IAM from VSAM already have regularly scheduled file reorganizations, which will generally be sufficient for IAM datasets. Depending on the dataset and application activity, the reorganizations may be done daily, weekly, monthly, or even quarterly.

Some of the symptoms of an excessively large usage of extended overflow are:

- Long elapsed time to open the dataset.
- Excessive use of virtual storage or the IAM Index Space.
- High I/O rates when processing the dataset sequentially.
- Potential inability to open the dataset or other datasets due to virtual storage constraints.

It can be difficult to predict the level of extended overflow usage at which serious performance deterioration will occur. For example, one file could have well over a million records in overflow, and not be experiencing any noticeable performance difficulties, whereas another dataset may only have a few hundred thousand records in overflow and be experiencing severe symptoms. The key factors are the key length and the general placement of records in the overflow area. For example, if a file has a key length of 4 with a million records in overflow, the storage used for that index is going to be substantially less than if the file had a key length of 64. If the records in overflow are in a generally ascending key sequence, or in clusters of ascending key sequence, then the I/O impact and processing time to open the dataset will most likely not be seriously impacted. A very random pattern of records through out overflow can have a serious impact on sequential I/O performance, and the processing time to open a dataset.

One of the cautions is for reorganizations that are done by application programs. Some application programs reorganizations are done by a single record load followed by a mass insert. This is not a valid reorganization from the access method point of view. The resulting dataset will frequently be in a less than optimum status after the application reorganization. If such a technique is used, the application reorganization should be followed by a file reorganization that is done by FDRREORG or an IDCAMS REPRO.

The other thing to watch out for on these scheduled reorganizations is where within the batch job stream they occur. For example, some applications reorganize a dataset after they are closed online, and then execute a large batch update process. The batch update process can result in heavy overflow use, so that when the dataset is subsequently opened for online processing, it is in a less than optimum state. By simply scheduling the reorganization after the update processing, the file will be in the best possible organizational state when it is opened for online processing.

GUIDELINES FOR REORGANIZING

Some guidelines for determining when an IAM dataset should be reorganized include the following:

- When more than 5 to 10 percent of the records in the dataset are in extended overflow.
- When the size of the Extended Overflow Index exceeds some storage quantity, such as 16 megabytes.
- When the Overflow area exceeds a quantity of DASD space, such as 1,000 cylinders.
- When a single volume dataset is approaching sixteen extents.
- When the number of overflow records for a particular dataset approaches or exceeds a predetermined number of records. IAM can assist in monitoring this if the file is defined with an Overflow (O=) override of that specified value.

For some of the above guidelines, IAM will issue an informational message, IAMW22, indicating that reorganization is recommended along with the reason. The IAMINFO reports will also include an IAM373 message indicating that reorganization is recommended. Several of these factors are available as selection criteria on FDRREORG, which provides an automatic method for reorganizing files only when needed. Other methods of automating file reorganizations include using the reports generated from the IAM SMF records, by IAMSMF or IAMSMFVS. Full information on IAM dataset reorganizations is provided in Section 10.81 of the IAM Manual.

10.50 CONTINUED . . .

LARGE PRIME INDEX

Datasets that have a prime index structure that exceed a few megabytes are considered to have a large prime index. The amount of storage required for the Prime Index, and whether or not it is compressed, is provided in both the IAMINFO reports and the IAMPRINT reports. Having a large prime index structure will not necessarily cause any type of performance problem, however such files may realize improved performance by reducing the prime index size. There are a number of factors to consider. The potential advantages of reducing the prime index size include faster index search time and reduced virtual storage requirements. The reduction in virtual storage may be partially, or in some cases entirely, offset by an increase in buffer size when the block size is increased. The costs are increased search time for records within each data block and increased physical I/O time.

The prime index size is based on the number of prime blocks, the key length, and the compressibility of the key structure. From a tuning perspective, the one thing that you may have some control over is the number of prime blocks. The first thing to make sure of is that the data compression is enabled for the dataset. This can help reduce the number of prime blocks by fitting more data within each prime data block. The next factor to check is for an excessively large CI freespace area. Make sure that such a large CI freespace area is warranted based on insert or record growth activity to avoid overflow growth. Reducing an excessively large CI freespace will result in fewer prime blocks. Increasing the block size for a dataset with a large CI freespace area may not be very beneficial. Next, if the file is at less than _ track blocking, increasing the block size will reduce the prime index storage. Changing the block size requires some caution, unless the dataset is quite predominately sequentially processed. Random processing, or short sequential browses that are typical of online systems may incur increased response times when using a larger block size, due to the increase in data transfer time. Plus they are also subject to increased CPU time to search the data block for the required record.

WHEN TO INCREASE THE BLOCK SIZE

So, when is it beneficial to increase the block size? There are two factors to consider. The first is the average record size as the data is stored, and the second is the benefit of buffering. As record sizes increase, there will be more benefit to increasing the block size providing that buffering is reducing physical I/O. The average stored record size is provided on the IAMINFO report for the file load. If that is not readily available, then the approximate value can be calculated with data from an IAMINFO or IAMPRINT report as follows:

$$\frac{\text{Blocksize} * ((100 - \text{CI Freespace}) / 100)}{(\text{Total Records} - \text{Inserted Records} + \text{Deleted Records}) / (\text{Number IAM Data Blocks} - 2)}$$

Figure 27: Calculation for Approximate Average Record Length

The benefit of buffering can be easily determined from data in the IAMINFO report by dividing the Disk Blocks Read by Requests Processed. This presumes of course that an adequate number of buffers are being provided. As this percentage of requests requiring I/O gets smaller, the benefit of buffering is increasing. The larger the benefit of buffering, the more likely it is to achieve benefit by increasing the block size. There may not be much benefit, from the physical I/O perspective if more than 50% of the logical requests require I/O. The I/O benefit is likely to be larger as the percentage drops to 25%, 10%, or even lower.

10.50 CONTINUED . . .

HOW TO INCREASE THE BLOCK SIZE

As a general rule, if the average record size is 1K or more (1024 bytes), and there is some beneficial buffering, there should be no hesitancy about increasing the block size. The block size, or blocking factor, can be changed by either using the CREATE IAM Override B= during the file define, load, or reorganization to specify a block factor. The alternative is to increase the CI size on the Define statements. For example, specifying a B=2 override will force half-track blocking. A blocking factor of 1 is not recommended on most current DASD devices because there will be a considerable amount of DASD space wasted due to the limitation of the IAM dataset block size of 32K.

```
//IAMOVRID DD  *
       CREATE DD=&ALLDD,B=2
/*
```

Figure 28: Example of IAM Override to set 1/2 Track Blocking

For datasets with smaller average record sizes, increasing the block size can be considered and will be beneficial with larger prime index structures as long as there has been beneficial buffering. There probably is not much benefit to increase the block size for files with average record sizes of less than 500 bytes, unless the I/O activity is predominately sequential, or there is a severe virtual storage constraint, which is discussed in Section 10.51.

HIGH I/O RATES

This section will discuss some of the common causes of higher than expected physical I/O's, commonly referred to as EXCP count. The IAMINFO report is a necessity to understand such a problem. The key statistical fields from the IAMINFO report that are used include the following:

- **DISK BLOCKS READ:** The number of physical I/O's (EXCP's) that were issued to read data from the IAM dataset.
- **DISK BLOCKS WRITTEN:** The number of physical I/O's (EXCP's) that were issued to write data to the IAM dataset.
- **SEQ CHAINED BLOCKS READ:** The number of additional data blocks read in as part of a sequential I/O. This number plus the **DISK BLOCKS READ** is the total number of blocks read into storage.
- **SEQ CHAINED BLOCKS WRITTEN:** The number of additional data blocks written out as part of a sequential I/O. This number plus the **DISK BLOCKS WRITTEN** is the total number of blocks written out to DASD.

The total EXCP count for the IAM dataset can be easily calculated by adding the **DISK BLOCKS READ** and **DISK BLOCKS WRITTEN** values. It is quite useful to have the two separate values, as they will help in our search for what is going on with the dataset. Some of the circumstances and potential actions are described below.

If the value for Disk Blocks Written is very high, then most likely what is happening is IAM is not deferring the writes for random updates. This situation occurs when the dataset is defined with Share Option 3 or when a Share Option 1 or 2 dataset is processed asynchronously, as is done by CICS. For online systems, this generally is a desired action so no change is recommended. For datasets defined with Share Option 3, they can be redefined with Share Option 2 because of the very high risk associated with sharing an IAM dataset for update.

If both the Disk Blocks Written and Disk Blocks Read are very high, such that they equal or exceed the total requests, the most likely cause is that the file is defined with Share Option 4, unless IAM RLS is active for this dataset. Setting Share Option 4 forces IAM to use only 1 buffer, and IAM will always reread a data block whenever it is requested, even if it is already in the buffer. Plus, IAM will always immediately write out any updated data block, including sequentially updated data blocks. The dataset should be redefined with a Share Option of 2, because sharing an IAM dataset for update is most likely going to result in a corrupted dataset, and lost data.

10.50 CONTINUED . . .

HIGH I/O RATES (Continued)

If both Disk Blocks Read and Seq Chained Blocks Read are exceedingly high, the problem is most likely that IAM is rereading empty prime or PE blocks. This can result due to an application having mass deleted a large group of records that occupied contiguous blocks, followed by attempts to retrieve records using a key greater or equal type of search. Depending on the Share Options and how the dataset was opened, IAM is able to avoid this type of processing. The affected dataset should be reorganized to resolve the problem. Try using the REREADEMPTY=NO IAM ACCESS override which may prevent the high I/O rate.

If Disk Blocks Read is quite high for a basic sequential I/O type of job, then the most likely cause is that there are a lot of records in key sequence that are scattered through many different Extended Overflow blocks. Such a situation is also likely to be coupled with a long time to OPEN the dataset, due to the Extended Overflow index build process. The solution to this problem is to reorganize the dataset.

USING MULTIPLE VOLUMES FOR PERFORMANCE

For datasets that have an unusually high I/O activity and are not on a device with PAV (Parallel Access Volume), it may be quite beneficial to spread the dataset across multiple volumes. By so doing with Enhanced Format IAM datasets, there can be concurrent physical I/O scheduled to each DASD volume, which may result in significantly improved online response times. With a little bit of planning, this is easy to accomplish by setting up proper space allocation parameters. IAM does not support key ranges, so a dataset cannot be spread across volumes in that manner. Two different techniques for accomplishing this will be shown. For both examples, it has been determined that the dataset requires approximately 2,000 cylinders of space, excluding overflow requirements. The bulk of the dataset will be split across 4 DASD volumes, however a fifth volume will be used to handle any potential growth into the IAM Extended areas.

The first example can be used for installations that have DFSMS active on their system. Note that the dataset does not have to be SMS managed for this technique to work, just have to have DFSMS active. If the dataset is going to be on SMS managed volumes, then the dataset must be defined with Guaranteed Space. If the dataset is being allocated to non-SMS managed volumes, then IAM allocates the dataset as if it were being defined with Guaranteed Space under DFSMS. That is, IAM will allocate the primary space quantity on each volume when the dataset is defined. For this technique to work, the secondary space quantity must be 0, which will prevent the usage of secondary extents. File expansion is accommodated by utilization of the space on the fifth volume.

```
//DEFMULTV  EXEC PGM=IDCAMS
//SYSPRINT  DD   SYSOUT=*
//SYSIN     DD   *
  DEFINE CLUSTER          -
    (NAME(MY.IAM.KSD)     -
    OWNER($IAM)           -
    VOLUMES(MYVOL1 MYVOL2 MYVOL3 MYVOL4 MYVOL5)        -
    CYL(500)       RECORDSIZE(100 1000)                -
    KEYS(24 8)     FREESPACE(5 20)                     -
    SHAREOPTIONS(2 3)   REUSE  )
   LISTCAT ENT(MY.IAM.KSD) ALL
/*
```

Figure 29: Example of Spreading IAM Dataset across Multiple Volumes

10.50 CONTINUED . . .

USING MULTIPLE VOLUMES FOR PERFORMANCE (Continued)

In the next example, a different technique is used in that the dataset will be allowed to take secondary extents. This is effective for files that are not DFSMS Extended Format files. To achieve the desired split of 500 cylinders across 4 volumes, a primary of 200 cylinders is being requested, with a secondary of 20 cylinders. The secondary results in a total of 300 cylinders, being 15 extents times 20 cylinders. The IAM overrides of MAXSECONDARY=1 is specified to prevent IAM from increasing the secondary allocation, and an override of MULTIVOLUME=PRIMARY is specified to cause IAM to allocate the primary on the next volume for the first extent.

```
//DEFMULTV EXEC PGM=IDCAMS
//SYSPRINT  DD  SYSOUT=*
//IAMOVRID  DD  *
   CREATE   DD=&ALLDD,MAXSECONDARY=1,MULTIVOLUME=PRIMARY
/*
//SYSIN     DD  *
  DEFINE CLUSTER              -
    (NAME(MY.IAM.KSD)         -
    OWNER($IAM)               -
    VOLUMES(MYVOL1 MYVOL2 MYVOL3 MYVOL4 MYVOL5)         -
    CYL(200 20)   RECORDSIZE(100 1000)                  -
    KEYS(24 8)    FREESPACE(5 20)                       -
    SHAREOPTIONS(2 3)   REUSE  )
   LISTCAT ENT(MY.IAM.KSD) ALL
/*
```

Figure 30: Example of Spreading IAM Dataset Across Volumes

10.51 STORAGE USAGE

STORAGE OVERVIEW One of the frequent questions is how much storage does IAM use, and what can be done to reduce the virtual storage requirements. IAM does require virtual storage to provide the services requested by application programs. IAM uses virtual storage to reduce I/O and the CPU time required to process an indexed dataset. One of the features of IAM is that it keeps the index to the file in virtual storage. There are no index buffers, or index I/O after the file has been opened. Data buffers are the other major component of virtual storage usage. These large storage areas are always requested from the 31-bit addressable area of memory (above the line). This use of storage seldom is a problem for batch jobs. However, large online regions that have hundreds of files open at any point in time may run into virtual storage constraints. This section will attempt to explain IAM's storage usage, and what potential there is to reduce the virtual storage requirements.

VIRTUAL STORAGE MANAGEMENT FEATURES To help minimize the need for virtual and real storage tuning, IAM has several special features to aid in the dynamic management of virtual storage for jobs using Enhanced Format IAM datasets. IAM can put the index for an open IAM dataset into a data space, which is referred to as an Index Space. This feature alleviates virtual storage contention by moving a lot of IAM storage into it's own data space. IAM can also dynamically increase the above the line storage region based on values from the IAM Global Options Table. This provides for a quick way to dynamically adjust to unexpected storage requirements without having to change the IEFUSI exit or the job's REGION parameter. When acquiring a non-critical area of storage, such as acquiring an additional buffer, if the storage was acquired below the 16-megabyte line, IAM will release that storage. This way, once the above the line region is filled, IAM will not unnecessarily use below the line storage, which could quickly disappear. IAM also monitors buffer usage, as a part of the Real Time Tuning, and will release infrequently referenced buffers.

BASE STORAGE REQUIREMENTS The actual amount of storage used to process each file, excluding load modules is provided on the IAMINFO report. This includes the amount of the total storage required and the amount of that storage which was above the line. For the Enhanced format files, IAM will always allocate virtual storage in multiples of 4K size areas, and manage the allocation of that storage. This is done to help prevent storage fragmentation and to improve reliability by reducing the chances of storage corruption that can easily occur when multiple software products are sharing the same page of virtual storage.

To best explain the concern about storage fragmentation, an example may be helpful. In this example, we will assume that the IAM file has _ track block size on a 3390, which is 13682. Whenever MVS allocates storage for a request that is for an area size that requires multiple pages, virtual storage is always acquired by assigning a new set of contiguous available pages. If IAM were to issue the getmain for the exact block size, MVS will always acquire 4 pages, or 16K bytes on a page boundary. The unallocated portion of that storage will be available for other smaller storage requests. In our scenario, during open IAM typically will acquire four buffers of 13682, resulting in MVS using 64K of virtual storage, with fragments of a little over 2K free scattered in that 64K area. Now, it is quite likely that another program or perhaps IAM for another file if IAM did not round the size requested, will use portions of those free areas. When one or more of those buffers are released, MVS will release only 3 full pages, and the fourth page will remain allocated to the subpool. So, when the file is closed that results in freemaining the buffer storage, the result will be four 12K areas of free storage, with storage allocated to the subpool being interspersed. When the file is reopened and the buffers are reacquired, MVS will look for 4 contiguous pages for each buffer. The previous storage area that was used for the buffers cannot be used because of the 4K areas still allocated to the subpool; hence new 16K areas are allocated for each buffer. This can and has resulted in the effective loss of most of the original 64K used for the buffers. For an online system, where files can be opened and closed multiple times, this fragmentation will eventually result in running out of usable virtual storage. To help prevent this occurrence, IAM will always issue the getmain for each buffer rounded up to a 4K boundary. While some of that virtual storage will not be used, when IAM releases the buffers, the entire area will be available for reacquisition of the buffer. So, instead of using 128K of virtual storage after one closing and reopening the dataset, only 64K will be used.

10.51 CONTINUED . . .

BASE STORAGE REQUIREMENTS (Continuued)

The minimum storage requirement per open Enhanced format file is 20K, which is divided into five separate areas, not including the index, buffers, and load module storage. The typical average for most moderately sized datasets is probably in the range of 28K to 32K. Amongst the variables that can cause an increase in the base storage requirements are:

- Need for a Data Compression work area, size is calculated as follows:
  (Maximum Record Length * 2) + 1024.
- Need for an Index Compression work area, size is calculated as follows:
  (4 * Key Length) + 1024
- The maximum number of buffers (MAXBUFNO)
- The number of VSAM Strings requested (STRNO), the size of each PLH is calculated:
  (Key Length * 2) + 552
- Maximum Record Length and Key Length
- Number of prime blocks with records in Extended Overflow

BELOW THE LINE STORAGE

IAM limits usage of 24-bit addressable memory (below the line) as much as possible. IAM generally requires only 4K of storage below the line to handle the I/O control blocks and channel programs. This amount may be larger if more IOB's are needed, or if the file has a very large number of extents. Note that the initial number of IOB's obtained is based on the STRNO value provided.

IAM requires approximately 4K of virtual storage to hold the simulated VSAM control block structure, which may reside either above or below the line, depending on what was specified in the ACB. The base VSAM control block area is 2352 bytes for a KSDS type of file, or 752 bytes for an ESDS plus the storage required for each string, or place holder. This area can also exceed 4K, if a larger value is specified for STRNO, which indicates the number of place holders. The place holder size is also impacted by the key length, as described above. For CICS, the VSAM control block area is above the line in 31-bit addressable storage.

ABOVE THE LINE STORAGE

IAM keeps all the rest of the required control information, buffers, and index above the line. The base IAM control block area requires 4K. IAM has a buffer table that will fit within an additional 4K as long as the MAXBUFNO value does not exceed 128. There is the prime block to overflow table, which has a minimum size of 4K, and may be larger depending on the file size. There are some work areas for data compression, index decompression, and the high level extended index which depending on their size requirements may fit within the other IAM storage areas.

BUFFER STORAGE

The buffer storage is broken down into single block areas, the size of each is the block size rounded up to a 4K value. A buffer for a typical _ track blocked IAM dataset on a 3390 type of device requires 16K. The maximum buffer storage used is easily calculated by taking the buffer size value, and multiplying it by the maximum number of buffers used from an IAMINFO report. Whenever IAM is acquiring a buffer, if the storage assigned is below the 16-megabyte line, IAM will release the storage.

INDEX STORAGE

Three are three different index areas for an IAM dataset. The first is the prime index, which is created when the file was loaded. The second is the index for the Extended PE area, and the third is for the Extended Overflow area. Both the second and third index areas are dynamic, and will change as the file is updated. The Extended PE index is saved within the dataset, but the Extended Overflow index is always dynamically built during open processing.

10.51 CONTINUED . . .

PRIME INDEX The prime area index is fixed in size at completion of the file load or reorganization, and will never be updated. This index is based on the high key in each prime block. IAM provides the capability to compress the prime index in a proprietary format that can greatly reduce the amount of storage required for the index. Index compression is an automatic feature that will be used whenever the prime index exceeds 8000 bytes, and the attributes of the key fit within the compression criteria. The amount of storage required for the prime index is provided in the IAMINFO report and on the LISTCAT IAMPRINT report. Take the indicated value from one of those reports, and round it up to 4K to determine the amount of storage that will be used for the prime index.

The storage for the prime index will come out of either the extended private area of the job opening the dataset, or optionally in a data space created by IAM just for the particular job step, which is called an Index Space. The Index Space can be activated either through the IAM override facility, or set in the IAM Global Options Table. By default, the prime index will reside in an Index Space for files that are opened under CICS.

EXTENDED PE INDEX The next index storage area is for the Extended PE blocks. This index, like the Prime Index, is based on utilizing the high key within each data block. The size of each entry is the key length plus four bytes, with an entry for each Extended PE block. Due to the internal structure of the Extended PE index, which is organized based on an internal grouping of the extended index blocks, the total storage used for the Extended PE is difficult to predict. Files with large quantities of Extended PE blocks, which are clustered together may not necessarily use any more storage than a file with a few Extended PE blocks that are sporadically space throughout the Extended area of the file. While this index is not compressed, it is still a relatively efficient format especially because only a very few files actually have need for this index. The Extended PE index will also use the Index Space, if one is available.

EXTENDED OVERFLOW INDEX The last segment of the index storage is for the Extended Overflow blocks. This is a record based index structure, consisting of an entry for each record in overflow. This index is subset into smaller groups, where each grouping consists of the overflow records from a particular prime block. An overflow index search is only done once it has been determined that the prime (or Extended PE) block that should contain the record has associated overflow records. This type of structure is expected to reduce the number of overflow index searches, reduce the number of entries any single search has to scan, and reduce the IAM CPU time for many functions related with overflow.

Estimating the actual storage requirements for the overflow index is difficult. The entries within each subset have compressed key format, but each subset also has header information. Each subset may have some empty entries. As a rough estimate, add four to the key length, and multiply that by the number of records in overflow. The result is the size of the overflow index prior to compressing, which may reduce the storage requirement, although the headers for the subsets will increase the storage requirement.

The Overflow index will reside in the Index Space when available, otherwise by default it will reside in Extended Private area of virtual storage.

10.51 CONTINUED . . .

INDEX SPACE

As discussed above, IAM can place the Prime Index, the Extended PE Index, and the Extended Overflow Index in a data space. By relocating these potentially large index areas into a data space, there is more virtual storage available within the job step region. This is expected to be of benefit to large online regions, which may have several large IAM files open. This data space is dynamically created by IAM when the first file is opened that will be using the Index Space, and is retained until the job step terminates. For any job step, there will be only one IAM Index Space, with all open IAM files using the same Index Space.

By default, IAM will only use an Index Space when running under CICS. This can be changed by either changing the IAM Global Options, or on a job step and file by file basis with the IAM ACCESS Override INDEXSPACE. The reasoning behind the default is that using the Index Space does cause a small increase in the CPU time for IAM processing. However, the Index Space is really of benefit to large CICS regions, which may have a very large number of open IAM datasets, requiring lots of virtual storage. So, to avoid the potential increase in CPU time when there may be no benefit of using the Index Space, we have chosen to set the default to use the Index Space for CICS. Customers that have many large IAM files may find it advantageous to change the Global Option to allow Index Space usage for batch jobs as well.

The size of the data space requested for the Index Space is taken from the IAM Global Options Table, using the value specified for DATASPACE. Note that this is the same value that is used for the data space obtained for a file load. The Index Space is created to be extendable, with the maximum size set to four times the DATASPACE value. The default value for the data space size is 256 megabytes. The IAMINFO report has been enhanced to include information on the data space usage. This includes the data space storage required for the particular dataset, in addition to the total data space usage for the job. By monitoring these values, you can determine if the default data space size has to be increased.

REDUCING IAM STORAGE USAGE

Now that we have an understanding as to how IAM uses storage, we can look at some of the different ways to reduce storage use. Certainly the easiest thing to do is to reduce the number of IAM buffers. For the best results, resist that temptation, and do the homework. With IAM's Real Time Tuning capabilities, IAM generally does an excellent job at buffer management, dynamically adjusting the number of buffers being used, to minimize delay and optimize resource usage. Frequently, the problem is more with the amount of storage used for the various index pieces, rather than with the number of buffers.

The first step is to make sure that you have adequate data on which to base your decisions. The IAMINFO reports from several days should provide the necessary information. Other data may be necessary to determine the whole virtual storage picture for the job or CICS region. The second key factor is why do you need to reduce the virtual storage being used. Some of the typical reasons are being unable to open IAM files due to insufficient storage, insufficient virtual storage left for application or system use, or to reduce the amount of paging being done.

Using the Index Space feature of IAM will significantly help reduce virtual storage contention within a CICS or batch region, and should resolve most of the typical storage problems encountered. One of the things that might easily be missed when converting VSAM files to IAM is reducing the number of buffers for the VSAM LSR buffer pools. As a usability feature, Enhanced format IAM datasets can be opened while still residing in a VSAM LSR buffer pool, however IAM does not use any of the buffers in the LSR pool. The number of buffers in the LSR buffer pool should be reduced by the number that are typically used for the converted files. Failure to do this could also cause a problem with paging, if the LSR buffer pool(s) are shared with other files that have not been converted to IAM. If the size of the pool remains unchanged, now more buffers are available to the other files, which may be significantly more than needed to meet response time requirements, resulting in increased real storage and virtual storage contention.

The two areas in which it may be possible to reduce the storage requirements for IAM files are buffers and index. With IAM's Real Time Tuning that includes dynamic buffer adjustment, reducing buffers will generally have an adverse impact on overall performance for the dataset, and buffers frequently are not the major area of storage problems. In most situations, it is the index areas that utilize the bulk of the virtual storage. So, we will start by looking at some things that might be able to be done to reduce the storage requirements for the index.

10.51 CONTINUED . . .

REDUCING PRIME INDEX SIZE

The Prime Index and the Extended PE index are based on the high key in the each block. These index structures can be reduced in size by reducing the number of blocks in the prime and Extended PE area of the file. The best way to accomplish that goal is to use IAM Data Compression on the file. This reduces the number of blocks by providing the ability to store more data records within each individual block, assuming that data compression is effective for the data within this file. This may also provide the additional benefit of reducing physical I/O (EXCP's) to the dataset, because there is more data within a block. If you are already using IAM data compression, you may want to consider trying the hardware compression, as that may achieve greater compression than IAM software compression.

Another alternative for reducing the number of prime blocks is to increase the block size. This is a beneficial approach for very large files with large record lengths. For example, if a file is using the typical _ track block size, the prime index storage can be cut approximately in half by using _ track blocking. Care must be taken when considering this alternative. The general rule of thumb is to only consider files that have at least an average record length of 500, and that require at least 1024K of storage for the prime index. There are several reasons to proceed down this path with some caution. First, for jobs or online systems that do a lot of random processing, increasing the block size but keeping the same number of buffers will double the virtual storage requirements for the buffers. The same number of buffers may be required to achieve the desired I/O performance. Also, the actual physical I/O performance will be affected due to the longer data transfer time of the larger blocks of data. The third potential problem is also for random types of requests, where searching each data block for the desired record will use more CPU time than with the smaller block sizes. These factors have to be considered, along with the benefits of reducing the virtual storage for the index.

For example, if you have a file that is using eight megabytes of storage for the prime index, with a record size for many records being in the 1,000 byte range, there should be no hesitation to use _ track blocking on that file. However, if a file has 512K storage requirement for the prime index, and an average record length of 50, then increasing the block size could be quite detrimental to performance, particularly if there is a large volume of random I/O activity for that file.

REDUCING EXTENDED OVERFLOW INDEX SIZE

Because the Extended Overflow Index is a record based index structure, the only way to reduce the size of the index is to have fewer records in it. This is normally accomplished by performing file reorganizations. While there may be a need for more frequent reorganizations, the timing of the reorganizations can also play a role. For example, it is not uncommon for installations after closing a file for online usage, to perform a file reorganization that is followed by batch job that does mass updates. After the mass update, which may have added a lot of records into overflow, the file is then reopened for online processing with a large overflow index. By changing the scenario to perform the reorganization after the mass update, the result will be that the file can be opened online without that massive overflow index structure.

One thing to watch out for are file reorganizations that are done by application programs. Some times such reorganizations do not result in a reorganized file from the file structure point of view. This is because they may do a single record load, followed by a mass insert. Such file reorganizations result in the file consisting entirely of Extended PE and Extended Overflow records. Some COBOL programs may be inadvertently doing this when they open a file for OUTPUT with ACCESS IS DYNAMIC or ACCESS IS RANDOM. If your application performs this type of file initialization, then it is highly recommended that you reorganize the file with FDRREORG or an IDCAMS REPRO after it has been loaded through such a mechanism, to reduce virtual storage requirements for the index structure.

If regularly scheduled reorganizations are not part of the production job streams, then the FDRREORG product offered from Innovation can be used to automate the file reorganization when needed, using various criteria. Other automation ideas would be to write programs to read the IAM SMF records, or to read the output of the IAMSMF or IAMSMFVS programs.

The other way to reduce the Extended Overflow Index is to prevent records from going into the Extended Overflow area to begin with. This can be a difficult task to accomplish. In some cases, it can however be as easy as increasing the CI freespace, which can particularly be successful if the insert records are distributed across the entire key range of the file. This may require some experimentation to determine what free space values will provide the file with the most benefit. Another way to reduce records in overflow is to preload the file with dummy records with key values similar to expected inserts, then delete the dummy records leaving empty space for file growth.

10.51 CONTINUED . . .

REDUCING IAM BUFFERS

In general, unless there is a problem with real storage contention, the IAM buffering requires no adjustment, except perhaps to increase MAXBUFNO for heavily accessed files. Some symptoms of real storage contention include:

- Increasing transaction response times even though I/O and CPU times have decreased
- Increased working set size for online system
- Increased demand paging
- Increased paging and / or swapping on host system
- Overall deterioration of system through put

When there is a real storage constraint, it may be necessary to impose tighter restrictions on IAM buffering. The intent of this section is to provide some guidelines for reducing IAM buffers when there is a storage constraint in online systems. This same technique can be used to reduce buffers for a virtual storage constraint, although reducing buffers should only be done in that circumstance as a last resort. It would be preferable to increase the region for the affected jobs rather than reducing buffering, as long as the system has none of the symptoms of a real storage problem cited above.

The general approach is to select general criteria of MINBUFNO and MAXBUFNO for most files, and then provide some specific higher values for the heaviest accessed files. Before adjusting the buffering, the storage analysis described above should be done, and make whatever adjustments are possible to reduce index sizes. Then after making the reductions in index storage requirements, obtain a new set of IAMINFO reports to consider buffer adjustments.

The first step in this approach is to sort the IAMINFO reports, in descending order of requests processed. Normally, files can be placed into three categories of heavily accessed, moderately accessed, and lightly accessed. From the information available on the IAMINFO reports, the value of the buffers needs to be determined. To aid in that understanding, an explanation of some of the fields in the IAMINFO reports may help.

- **DISK BLOCKS READ:** This figure represents the number of read EXCP's that were issued. If sequential chaining has occurred, there may have been multiple blocks read per each EXCP, although this is rare in an online environment.
- **DISK BLOCKS WRITTEN:** This figure represents the number of write EXCP's that were issued. For an online environment, and for batch jobs with SHAREOPTIONS of 3 or 4, IAM always immediately writes out any randomly updated block.
- **DYNAMIC BUFFER RETRIEVALS:** This figure represents the number of read I/O's that were avoided due to IAM using more than one buffer. This is the raw number representing the I/O savings of having multiple buffers.

In determining the value of buffers, the critical question is whether or not the I/O savings is worth the additional storage requirements. To help determine that, there are two key indicators that are useful. These key indicators are approximations, and certainly do not reveal the entire picture of what is occurring. The underlying assumptions that the file activity for these files is proportional to overall online activity may be entirely false. However, for this level of tuning they should be sufficient for the vast majority of applications. It is important to have some understanding of application activity, because it does have a bearing on tuning decisions. Certain files may have the bulk of their activity occur during start up, or at end of day when storage resources are not constrained. Tuning buffers for those types of files will have almost no impact on the bulk of the daily load. File activity profiles over a several hour period are not revealed by the IAMINFO reports.

10.51 CONTINUED . . .

CALCULATING PERCENT OF I/O SAVED

The first key indicator is what percentage of the READ I/O's is actually being saved. That is calculated by the following formula:

$$\frac{\text{DYNAMIC BUFFER RETRIEVALS}}{\text{(DISK BLOCKS READ + DYNAMIC BUFFER RETRIEVALS)}}$$

Figure 31: Calculating Percentage I/O Saved

The higher this value is, the more effective use of buffers is being made. Low values indicate less effective use of storage in a constrained environment. The implications of low values are that a lot of storage is being used to hold highly transient data, which has a low probability of being reused. Certainly physical I/O's are being saved, which is a key objective of IAM, however the price may be too high.

CALCULATING BUFFER RETRIEVALS PER MINUTE

The second key indicator, although it is admittedly a gross approximation, is the number of I/O's per minute that have been saved. The intent is to determine approximately whether the data in the buffers are being reused before the page(s) containing the buffers are paged out, and also to include the I/O activity rate in buffer considerations. To calculate this figure, use an approximation of the length of time that most activity occurs. For example, if the online system is up for twelve to sixteen hours, typically the bulk of the activity occurs during a seven or eight hour period. So, for this level of tuning, look at the I/O counts as if they occurred during an eight-hour period instead of the total length of time that the online system was active. Take the Dynamic Buffer Retrievals and divide it by the primary active period in minutes, resulting in buffer retrievals per minute.

A lot of judgment, coupled with the knowledge about file activity, is required to decide where buffers should be reduced. In certain cases, it may be very clear that buffers can be reduced without a significant impact. For other files, it is more difficult. It helps to have the key figures for each file as a line or column on a single sheet, so that comparisons can be easily made. The priority for reducing files should be on the files that have the least benefit from the buffers, while trying to retain or perhaps increase buffers for files that are receiving the most benefit.

BUFFER ANALYSIS EXAMPLE

As a starting point, if read I/O's saved is less than 25% and buffer retrievals per minute is less than the maximum number of buffers, then reduce the number of buffers. As a further criteria, if the buffer retrievals per minute is less than half the Maximum number of buffers, even with a high percentage of I/O requests saved, then reduce buffers. If the decision is made to reduce buffers, set MAXBUFNO to the buffer retrievals per minute. If there is a severe storage constraint, then more stringent criteria needs to be used. If there is only a slight storage constraint, then less stringent criteria can be used. A few examples may help. In the figures presented, assume that the bulk of the activity occurs during an eight-hour period.

	FILE 1	FILE 2	FILE 3
DISK BLOCKS READ	4930	3038	2389
BUFFER RETRIEVALS	11297	6736	1625
MAXIMUM BUFFERS	22	21	15
% I/O Saved	69.6	68.9	40.5
Buff. Retrievals / Min	23.5	14.0	3.4

Figure 32: Example Data for Buffer Analysis

10.51 CONTINUED . . .

FILE 1 ANALYSIS File 1 is making very effective use of the buffers, plus the activity rate is high enough that the probability is excellent that the required pages will remain in storage, unless the system is seriously constrained. Even though an IAM368 message indicating more buffers could have been used did appear, the activity level does not seem high enough to warrant additional buffers unless storage constraints are relieved.

FILE 2 ANALYSIS This file is also making effective use of the buffers, but has a lower activity rate than File 1. The decision on buffers for this file clearly requires some judgment, and weighing how much storage savings is needed versus the savings from I/O. While references to these buffers are probably not causing page faults, the storage tied up for these buffers may be better put to use for other storage needs. If this file's activity rate is moderate when compared to other files, then buffers should be reduced. However if this file's activity rate is relatively high compared to the other files, then buffers should be left as is, or only reduced by one or two.

FILE 3 ANALYSIS This file has beneficial savings in I/O's from the buffers, however the buffer retrieval rate is very low. Unless a system has no memory constraints, the MAXBUFNO should be reduced for this file. It seems almost to be certain that most buffer retrievals will result in page faults. As a starting point, reducing MAXBUFNO to six or seven seems reasonable, unless there is a big storage constraint problem, in which case lowering MAXBUFNO even more would seem reasonable. This file may be considered a candidate for the base buffer range selected for the light to moderate activity files.

BUFFERS FOR LIGHT AND MODERATELY ACCESSED FILES Lightly accessed files have very low file activity, and seldom use more buffers than were originally obtained at open, which is generally four. Almost every buffer access, including I/O requests result in page faults for these buffers. Moderately used files have file activity rates that may increase the buffers to the eight to twelve range. Once again, in storage constrained systems access to those buffers typically result in page faults. In fact moderate activity files may be more of a problem than light activity files, because IAM will select the least recently used buffer for input I/O, which is the buffer that is most likely to be paged out. If the file only had a couple of buffers, then the likelihood of it being in storage is better.

Generally, the recommendation is to set MINBUFNO=2 for light and moderate activity files. The setting for MAXBUFNO depends on how active the files are, and how effectively they use buffers. Generally, the recommendation is to set MAXBUFNO at 8, or perhaps down to 6. For seriously constrained systems, using 4 or 5 may be appropriate.

IMPLEMENTIN G BUFFER VALUES IAM offers several methods to change the buffer ranges. These include changing the Global Options Table, providing the MINBUFNO and MAXBUFNO overrides when file is defined, or providing the overrides for CICS. The recommended way is to provide an override for CICS, so that batch processing will be able to use buffers to their full advantage. The overrides can be used even if the IAM files are dynamically allocated by CICS. The CICS override method involves adding an IAMOVRID DD card to the CICS JCL, specifying a card image file. The file normally is only read once, when the first IAM file is opened. With the specification of REREAD on the override cards, the override cards will be read for each IAM file OPEN. This provides the capability to change the override values without having to bring down CICS for the new values to be read. Otherwise, any changes to the overrides after an IAM file is opened will not be picked up until CICS is restarted.

```
Example of IAMOVRID DD card:

//IAMOVRID  DD  DISP=SHR,DSN=cics.prod.cntl(IAMOVRID)

Example of contents of member IAMOVRID in cics.prod.cntl:

  ACCESS  DD=FILE1,REREAD,MAXBUFNO=24
  ACCESS  DD=FILE2,REREAD,MAXBUFNO=15
  ACCESS  DD=FILE3,REREAD,MINBUFNO=2,MAXBUFNO=4
```

Figure 33: Example of IAM Overrides to Change Buffering for Storage Constrained Systems

10.51 CONTINUED . . .

LOAD MODULE STORAGE REQUIREMENTS

Shown below is a list of the modules required for accessing Enhanced Format IAM datasets, with their approximate virtual storage requirements. Only one copy of each module is loaded, as required, regardless of the number of IAM datasets opened by a task. Most of these IAM modules are reentrant, and can be placed in LPA, although for ease of maintainability, that is not recommended. The two most frequent problems encountered with placing IAM in LPA are that an ineligible module is placed in LPA, and sometimes a customer forgets that IAM modules were placed in LPA, so they run into problems when installing a new version or maintenance level of the product. If there is a significant amount of IAM usage, then placing these modules in LPA may be of benefit, at least to the extent that it will reduce the time to load the modules when an IAM file is opened, and it will reduce private area storage requirements.

Module Name	Storage Required	RMODE	LPA Eligible	Description
IAMABUFR	24K	ANY	YES	IAM buffer manager and physical I/O driver.
IAMACCIX	13K	ANY	YES	IAM AIX and Path Logical I/O Request Handler
IAMACCKS	60K	ANY	YES	IAM Logical I/O Request Handler.
IAMACCXM	15K	ANY	YES	IAM Logical I/O Request Handler for IAM RLS
IAMADNAC	1K	ANY	NO	IAM Anchor.
IAMAPTOC	18K	ANY	YES	IAM Path and AIX Open and Close
IAMAVSOC	48K	ANY	YES	IAM Open, Close and support subroutines.
IAMAVS24	3K	24	YES	IAM interface to application program and user exits.
IAMAXTND	12K	ANY	YES	IAM routine to acquire an extent. Loaded on an as needed basis.
IAMCOMPH	2K	ANY	YES	IAM hardware compression routine, loaded only when using hardware compression.
IAMCOMPO	7K	ANY	YES	IAM software Data Compression Routine is loaded only when the IAM file(s) are data compressed.
IAMCRTVS	41K	24	YES	IAM File load processor is only loaded when IAM files are being loaded or reorganized.
IAMNINFO	22K	24	YES	IAMINFO Report Generator is loaded during close processing only if there is an IAMINFO DD card.
IAMOPT	2K	24	NO	IAM Global Options Table is not recommended for LPA.
IAMOVRID	13K	24	NO	IAM Override Processor for Compatible format files, is only loaded when there is an IAMOVRID DD card and Compatible Format files. Acquires a 24K table in above the line storage to hold the overrides.
IAMOVRIX	12K	24	NO	IAM Override Processor used for Enhanced Format files. Will only be loaded when there is an IAMOVRID DD card and Enhanced format files are opened. Acquires a 48K table in above the line storage to hold the overrides.

Figure 34: Table of IAM Load Module Storage Requirements

For processing typical data compressed Enhanced format files, without any IAM overrides, there is a requirement of approximately 5K of below the line storage, and 118K above the line for the IAM load modules. If there are overrides, then the below the line storage will increase to 16K, and above the line storage to 166K. Additional storage will be necessary during file open, close, and extend processing.

10.60 USING IAM ALTERNATE INDEX SUPPORT

OVERVIEW The optional IAM Alternate Index support allows you to define and use IAM Alternate Indexes and Paths to Enhanced Format IAM KSDS or ESDS type of files. As with standard IAM support, there are no changes required to application programs and generally no changes to JCL, other than to indicate on the definition that the base cluster is to be an IAM dataset. The DEFINE ALTERNATEINDEX and DEFINE PATH can be changed similarly, but there is no requirement to do so. All the rest of the alternate index processing is essentially the same as is being done with the VSAM alternate indexes today. Some changes may be required when renaming or creating copies with different names, to synchronize all the related datasets with the new names.

ALTERNATE INDEX An alternate index provides an additional index to an indexed type of dataset (KSDS), or an index to an entry sequence dataset (ESDS). Users can define one or more alternate indexes to any base dataset. The alternate index dataset itself is an IAM KSDS enhanced format type of dataset that is indexed by the alternate key. The records in the alternate index contain some control information, the alternate key, and the primary key value for the record in the base dataset with the corresponding alternate key. An alternate index is defined as containing UNIQUE keys when there is only one base record with any specific alternate key. Alternate indexes can also be defined as containing NONUNIQUE keys, where any particular alternate key can be contained in multiple base records. As with VSAM, the alternate index dataset can be explicitly processed by programs without referencing the base cluster, or can be used to access the records in the base cluster. To use an alternate index dataset to access the records in the base cluster, a PATH must be defined.

An additional attribute of alternate index datasets is whether they are upgradeable. When an alternate index is defined with the UPGRADE attribute, IAM will automatically update the alternate index as needed whenever updates are made to the base cluster, either through the primary key or when accessed through an alternate key. Any alternate index defined with the NOUPGRADE attribute will not be automatically updated by IAM, and it is the application programs responsibility to ensure that the alternate index is updated in a manner to remain synchronized with the base dataset.

PATH A PATH provides the mechanism to access a base cluster through an alternate index. When a path is defined, one must provide the name of the path and the name of the related alternate index to be used to access the base cluster, or the base cluster to be processed whenever the path itself is referenced. The UPDATE or NOUPDATE attribute is specified for each path defined. When a path is defined with the NOUPDATE attribute, any upgradeable alternate indexes will not be updated automatically when this path is used to access a base cluster either directly or through an alternate index.

For VSAM datasets, a PATH is only a catalog entry that is quite similar to an ALIAS entry. With IAM, a PATH will become a one-track dataset containing the name of the related alternate index or base cluster and the UPDATE or NOUPDATE attribute of the PATH. A single track dataset was chosen instead of an ALIAS type of catalog entry because of concerns about there not being adequate dataset management utility program support for ALIAS entries, which might result in the loss of the entry.

ALTERNATE INDEX ASSOCIATIONS For VSAM datasets, the associated dataset(s) in an alternate index sphere are linked together by an association cell within the catalog entry for the dataset. Because IAM datasets are cataloged as non-VSAM datasets, there are no association cells within the non-VSAM catalog entry. For this reason, IAM saves the association information within each of the associated IAM datasets. The association information consists of the dataset name and type (Base, AIX, or Path) of dataset, and includes the update or upgrade attribute. Whenever a LISTCAT is requested (or an IAM ISPF display) for an IAM dataset that is part of an alternate index sphere, the association information will be included in the output. The information in the association cell is critical, because it will direct the processing during an open of a component of the sphere, and when deleting an alternate index sphere. Note that each type of dataset includes the associated information a self-describing association entry. The associations are always listed in the order that each was defined, which will always begin with the base cluster.

10.60 CONTINUED . . .

BASE CLUSTER ASSOCIATIONS For a base cluster, the associations will include in addition to itself, all of the alternate indexes that have been defined for that base cluster, and all of the paths that have the base cluster as it's path entry dataset. Not included in the base cluster associations are the paths that have the alternate indexes as path entries. A base cluster could have several associations. The number of associations is limited by the block size of the IAM dataset. For a typical _ track blocked IAM dataset, it can have a maximum of 204 associated datasets.

ALTERNATE INDEX ASSOCIATIONS For an alternate index, the associations will include the related base cluster, it's own self-entry, and any paths that have been defined with this particular alternate index as it's path entry.

PATH ASSOCIATIONS The path associations will consist of the base cluster, the path, and if the path entry is an alternate index, then the alternate index.

ASSOCIATION MAINTENANCE In general, dataset management utility software products treat IAM files as non-VSAM datasets. Therefore, these products are not going to know about the associations between these non-VSAM datasets, which will require some additional effort on the part of users. The main areas of concern are renaming one or more of the datasets in an alternate index sphere, or performing a copy or restore to a new name of such a dataset. After changing the name of one or more components of an IAM alternate index sphere, it will be necessary to perform a DEFINE RECATALOG on all of the components affected to insure that the dataset names are all properly reflected in the saved association information. IAM includes special processing on a DEFINE RECATALOG to perform the renames on the internal association data. Procedures for performing this recatalog process are described later in this section. In Version 8.0, there is support to erase the associations from the base cluster, if for example a base cluster has been copied or restored with a new name. To erase the associations, one can perform a DELETE AIX NOSCRATCH on the base cluster, which will clear out all related alternate index information.

USING AN IAM ALTERNATE INDEX The procedure for defining and using IAM datasets with alternate indexes is identical to the VSAM process. The only difference is to specify on the DEFINE that the base cluster is to be an IAM dataset. This specification is typically done by either adding an OWNER($IAM) parameter to the define parameters, or changing the name to include the $IAM literal. Most applications that are going to be using IAM for their alternate index already have the general procedures established. To convert to IAM, the only change needed is on the define step of the base cluster. An overview of the process for using an IAM alternate index include the following steps:

- Defining the base cluster.(Section 10.20)
- Loading the base cluster with data (Section 10.30)
- Defining the alternate index(es) (Section 10.61)
- Building the alternate index(es), usually with IDCAMS BLDINDEX command.(Section 10.62)
- Defining paths to the alternate index(es) and if used, paths to the base cluster (Section 10.63)
- Using Paths to access data (Section 10.64)
- Special Considerations with IAM Alternate Indexes (Section 10.65)

The first two steps of defining and loading the base cluster are described in prior sections the IAM manual. Details and examples of those steps will not be presented here. This section will describe the rest of the procedures for using IAM Alternate Indexes.

10.61 DEFINING AN IAM ALTERNATE INDEX

OVERVIEW The process of defining an IAM alternate index is virtually identical to defining a VSAM alternate index. IAM will automatically determine if the base cluster is an IAM dataset, so there is no need to specify an OWNER($IAM) on the DEFINE ALTERNATEINDEX. The definition of an alternate index is quite similar to the definition of any other KSDS type of file, with the major difference being the addition of the RELATE(base cluster name) parameter to identify the base cluster. The KEYS parameter will specify the key length of the alternate key, and the position of the alternate key in the base dataset record. The RECORDSIZE parameter specifies the average and maximum size of the records within the alternate index itself. The minimum size for this record will be (5 + alternate key length + base key length). When the base cluster is an ESDS, the base key length will be the RBA length of either 4 or 8.

Two additional alternate index definition parameters are **UNIQUEKEY | <u>NONUNIQUEKEY</u>** and **<u>UPGRADE</u> | NOUPGRADE**. The UNIQUEKEY and it's opposite NONUNIQUEKEY indicate whether or not there is only one record in the base cluster with any particular alternate key (UNIQUEKEY), or if an alternate key can have multiple records in the base cluster with that key value (NONUNIQUEKEY). The IDCAMS default is NONUNIQUEKEY. The UPGRADE parameter and the opposite, NOUPGRADE, indicate whether IAM will automatically update the alternate index whenever the base cluster is updated in a way that would change the alternate index. The IDCAMS default is UPGRADE.

As with VSAM, an SMS managed alternate index will be given the same STORCLASS and MGMTCLAS as the base cluster. The user can specify a DATACLAS if desired.

10.61 CONTINUED . . .

DEFINING AN IAM ALTERNATE INDEX WITH IDCAMS

IDCAMS ALTERNATE INDEX DEFINE The most common method of defining an alternate index is by using the IDCAMS utility program. The name of the base cluster must be included in the alternate index definition, specified by the RELATE parameter. When IAM intercepts an alternate index define, it will look at the base cluster to see if it s an IAM dataset, and if so an IAM alternate index will be defined. Specifying the OWNER($IAM) parameter will force the define process through IAM, which will fail if the base cluster is not an IAM dataset.

The IAM alternate index is an enhanced format IAM KSDS type of dataset, which resides on disk as a non-VSAM file type, with a DSORG of PS set. The data and index component of the alternate index all reside within the single dataset. IAM CREATE overrides can be used when defining an alternate index as described as described in sections 10.20 and 30.02 of this manual.

BASIC PARAMETERS There is required information that must be provided to define an IAM Alternate Index. Below are the IDCAMS keywords that can be used to provide that information. Please note that you cannot provide an SMS STORCLAS or MGMTCLAS value on the define of the Alternate Index. This is a VSAM restriction that the alternate index must be in the same STORCLAS and MGMTCLAS as the base cluster.

Essential Keyword	Description
ALTERNATEINDEX(....)	Identifies that an alternate index dataset is to be created.
NAME(dsname)	This is a required parameter. The entry name specified for the alternate index will be the name of the IAM dataset. The data and index component names are ignored.
RELATE(dsname)	Required parameter that specifies the name of the base cluster that this dataset will be an alternate index to. To have an IAM alternate index, the base cluster must be an IAM Enhanced Format KSDS or ESDS type of file.
KEYS(length offset)	Required parameter that specifies the length of the alternate key, and the relative position of the alternate key within the base cluster. For IAM, the maximum length of the alternate key is 249. Note that each record in the base cluster must be long enough to contain the alternate key. The IDCAMS default values are: 64,0.
RECORDSIZE (average ,maximum)	Specifies the average and maximum lengths of the records in the alternate index dataset. The minimum record length is 5 plus the alternate key length plus the length of the key for the base cluster. Note for ESDS files the base key length is 4 bytes, or if defined with extended addressability (XESDS), it will be 8 bytes. For an alternate index defined with the UNIQUE attribute, the maximum length will be equal to the minimum length record. For an alternate index defined with NONUNIQUE keys, the maximum length will be the based on the maximum number of NONUNIQUE keys there could be for any alternate key. The formula is (5 + alternate key length + (maximum number of base records with same alternate key * primary key length)). Spanned records will be automatically indicated when the maximum record size exceeds the control interval size. The maximum record size for an IAM file is approximately 8 megabytes. The VSAM architecture allows for up to 32,767 non-unique keys per each alternate key, which IAM abides by to insure full VSAM compatibility. IDCAMS default values are: average=4086 maximum=32600.

10.61 CONTINUED . . .

CYLINDERS(xx yy)
TRACKS(xx yy)
RECORDS(xx yy)
MEGABYTES(xx yy)
KILOBYTES(xx yy)
Required information that indicates the amount of DASD space to be allocated for the IAM dataset. The unit of allocation is based on the keyword specified. The first value provided indicates the amount of space to be allocated during the file definition process. For IAM files, the primary quantity MUST be available on the first volume, otherwise the request will fail. The second value, which is optional, indicates the amount of additional DASD space to request in case the primary quantity is insufficient. The secondary quantity will be used to acquire additional extents during the BLDINDEX or reorganizations. The secondary quantity will also be used to acquire additional space as needed to handle updates and inserts after the BLDINDEX.

Because IAM datasets that are not DFSMS extended format are non-VSAM, they are limited to 16 extents per volume, and a total maximum of 255 extents. The maximum extent size is 64K-1 tracks (65,535), or 4,369 cylinders per volume, which is also the maximum amount of space that can be used by IAM on any single volume.

VOLUMES(volser….)
Specifies the volume(s) on which IAM is to allocate the dataset. A maximum of 59 volumes can be specified for an IAM dataset, due to MVS TIOT entry size limitations.

10.61 CONTINUED . . .

OPTIONAL PARAMETERS While the following keywords are all optional, in various situations they may be required or quite beneficial. Some of the keywords listed here are not relevant to IAM files, and are presented for that reason. For ease of reference, they are presented in alphabetical sequence.

Optional Keyword	Description
BUFFERSPACE(bytes)	Specifies the maximum amount of virtual storage to be used for buffers for this dataset. IAM will use this value to calculate the effective MAXBUFNO for accessing the dataset, providing it does not go below the default value from the IAM Global Options Table.
CONTROLINTERVAL-SIZE(size)	For VSAM, this controls the logical and physical block size on DASD for the file. IAM will refer to this value when calculating the block size it is going to use for the dataset, generally increasing the value to one that will maximize utilization on the device type to which the dataset is allocated. IAM does store the specified CI SIZE, so that it is available information if the dataset is converted back to VSAM.
DATACLAS(dataclas)	For SMS installations, this parameter specifies the name of the SMS DATA CLASS construct, which provides the allocation attributes for the new dataset. Attributes from the DATACLAS will be used, unless otherwise explicitly specified on the DEFINE statement. Values provided by a Data Class include maximum record size, key length and key offset, space allocation values, free space, share options, CI size, and volume count.
FILE(ddname)	Optional keyword that specifies a DDNAME that allocates the volume(s) on which the IAM file is to be allocated. This keyword can also be used to relate a particular IAM CREATE Override statement with a matching DDN= specification.
FREESPACE(CI%, CA%)	Specifies the amount of space to be reserved for future inserts or updates when the file is being loaded. **CI%:** Specifies the amount, as a percentage, of space to be left available in each prime block of the IAM file. Unless a file is never updated and never has records added to it, some CI free space should be specified. This is of particular importance to data compressed files that are updated, because even if the application does not change the length, the stored record may end up being longer after compression. **CA%:** Controls how much DASD space is released at the end of a file load. Using _ of the specified percentage, a target amount of DASD space to be reserved for future expansion is computed. If the amount of available DASD space within the file extent(s) is equal to or less than the amount to be reserved, then no space is released. IAM will not acquire additional extents to meet the space reservation. If the amount of DASD space exceeds the reserved value, then the excess will be released.
KEYRANGES(low high)	IAM will not split the data into key ranges. However if specified, IAM will make the secondary allocation quantity equal to the primary. This is because VSAM would use the primary for each key range, which IAM is not able to do.

10.61 CONTINUED . . .

MODEL(datasetname) Specifies that the attributes of the dataset being defined are to be copied from an existing VSAM or IAM alternate index. This capability is only relevant for basic file characteristics, such as record size, key length and offset, space allocation values, volumes, and free space values. Any IAM Overrides are NOT picked up by the MODEL parameter. Likewise, any VSAM file attributes that are ignored by IAM are not available either, such as IMBED, SPEED, REPLICATE, KEYRANGES, etc.

If you are trying to define a VSAM file using an IAM file as a model, you MUST provide an OWNER parameter, with a value that does not contain $IAM.

NOTE: When using MODEL with ANYVOL, the SUBALLOCATION parameter must also be specified. Although it is ignored by IAM, it prevents IDCAMS from trying to allocate ANYVOL.

OWNER($IAM) Optional keyword that indicates that the alternate index is to be an IAM dataset. IAM will automatically assume that an alternate index is to be an IAM dataset when the base cluster is also an IAM dataset. If no value has been specified for OWNER, IAM will set it to $IAM in the catalog entry.

RECATALOG Is an optional keyword for existing IAM datasets to reestablish the catalog entry, and update internal alternate index relationship information. Requires the user to specify the dataset name, the volume(s) on which the dataset resides, and the name of the related base cluster. Recatalog is also used after renaming one or more components of an alternate index sphere, to reset the associated dataset names to the new names.

REUSE
NOREUSE Specifies whether the file being defined can be reloaded (or reorganized) without being redefined. IAM defaults to REUSE, which is that any IAM file can be reloaded without having to be deleted and redefined. To use this feature with IDCAMS REPRO, specify the REUSE keyword.

IAM does provide a Global Option, ENABLE=NOREUSE, which if set will cause IAM to honor the specification of REUSE or NOREUSE. If that Global Option has been set, then IAM will honor the NOREUSE setting just like VSAM. While quite rare, there are a few application programs that rely on the NOREUSE setting.

NOREUSE will not allow a file to be reloaded without being deleted and redefined. An exception to this is made if the program issuing the OPEN is FDRREORG, in which case it will be allowed. If any other attempt is made to do so, the OPEN will fail with a return code of 8, and the ACB error flag set to 232, or x'E8'.

10.61 CONTINUED . . .

SHAREOPTIONS (cross-region ,cross-system) Specifies the level of protection provided by the access method to prevent or allow sharing of data within the file. The protection mechanisms include the OS/390 ENQ service, and the internal IAM buffering techniques.

The first parameter specifies how a file can be shared in the same system (CPU). The second parameter specifies how a file is shared between systems.

NOTE: With IAM RLS, IAM supports automatic record level sharing within the same MVS image for concurrent users of the IAM dataset. This is an enhanced sharing facility, that by default may be automatically provided for cross region share options values of 3 or 4. When IAM RLS is not active, IAM supports the cross-region share options with the MVS ENQ service, the same as VSAM. IAM does not support the cross-system share options. IAM issues an ENQ with a major name (QNAME) of IAMENQ and the dataset name plus first volume serial as the minor name (RNAME). If you need to enforce ENQ protection cross-system then you must add the major name of IAMENQ to your CA-MIM or GRS control files or whatever ENQ control product you use.

Cross Region Share Option Values:

1. Any number of users for read **OR** one user for update. The file's structure, data integrity, and read integrity are fully preserved.
2. Any number of users for read **AND** one user for update. The file's structure and data integrity are fully preserved. If the file is currently opened for update, other users reading the file do not have read integrity. They may not be able to access inserted or updated records, if such records were added to the overflow areas of the file, without closing and reopening the dataset.
3. Any number of users for read or update and users are responsible for integrity. Updated blocks are immediately written back out to DASD. *Use of this share option for IAM files is strongly discouraged unless the file is shared using IAM RLS, or some other VSAM sharing software.* Due to the nature and structure of the index to the IAM overflow area, the data integrity of IAM files is compromised by use of this share option value.
4. Any number of users for read or update, and users are responsible for integrity. IAM will use only a single buffer, and each logical I/O request will cause the buffer to be refreshed, and subsequently rewritten if the record is updated. *Use of this share option for IAM files is strongly discouraged unless the file is shared using IAM RLS, or some other VSAM sharing software.* Due to the nature and structure of the index to the IAM overflow area, the data integrity of IAM files is compromised by use of this share option value.

SUBALLOCATION IAM files are always allocated as if they were unique clusters. However it may be necessary to specify this parameter when using the IAM non-specific allocation (ANYVOL) and the MODEL parameter. SUBALLOCATION will prevent IDCAMS from allocating the volumes indicated in the VOLUME parameter when the MODEL parameter is specified.

10.61 CONTINUED . . .

TO(date)
FOR(days)

Specifies the retention period for the file being defined. This parameter has the same meaning for an IAM file as a VSAM file. The expiration date is placed in the VTOC for the dataset, and in the catalog entry. The keyword PURGE must be specified on the DELETE to cause the file to be scratched.

TO(date) – gives the date in the form YYYYDDD (four or two digit year and three digit Julian date), through which the IAM file defined is to be date protected.

FOR(days) – gives the number of days up to 9998, through which the IAM file being defined is to be date protected. A value of 9999 results in permanent retention.

Default: Dataset is not date protected.

UNIQUE

This parameter has no relevance for IAM files, as IAM files are always unique.

However, the user should be aware that specification of this keyword results in IDCAMS allocating the specified volumes prior to issuing the actual define request. For this reason, it is recommended that this parameter not be specified for IAM files.

If the customer is using the IAM non-specific device allocation, i.e. ANYVOL, then this parameter must not be specified.

UNIQUEKEY |
<u>NONUNIQUEKEY</u>

Specifies whether more than one record can have the same alternate key. When UNIQUEKEY is specified, each alternate key can only index one data record. With NONUNIQUEKEY, each alternate key can index one or more data records. The number of non-unique keys that can be contained in a single record is limited by VSAM architecture to 32,767.

Default is NONUNIQUEKEY.

<u>UPGRADE</u> | NOUPGRADE

Specifies whether IAM will upgrade (update) the alternate index as the base cluster is modified.

UPGRADE indicates that IAM will automatically update the alternate index whenever records are updated, inserted, or deleted from the base cluster.

NOUPGRADE indicates that IAM will not automatically update the alternate index.

Default is UPGRADE.

10.61 CONTINUED . . .

EXAMPLE A: BASIC ALTERNATE INDEX DEFINE

Shown below is an example of using IDCAMS to define an alternate index for an IAM base cluster. For ease of conversion from VSAM to IAM, the need to supply an OWNER($IAM) in the define has been eliminated for alternate indexes and paths. When defining the alternate index, if the base cluster is a VSAM dataset, then the define is passed to VSAM, as long as OWNER($IAM) is not specified. If the base cluster is an IAM dataset, then IAM will automatically define the alternate index as an IAM dataset.

```
//AIXDEFIN   EXEC  PGM=IDCAMS
//SYSPRINT    DD   SYSOUT=*
//SYSIN       DD   *
       DEFINE ALTERNATEINDEX                   -
              (NAME(example.iam.aix)           -
              RELATE(example.iam.cluster)      -
              KEYS( len  offset)               -
              RECORDSIZE(avg   max)            -
              SHAREOPTIONS(2 3)                -
              FREESPACE( ci%  ca%)             -
              VOLUMES(myvol1)                  -
              CYLINDERS(primary secondary)     -
              NONUNIQUEKEY                     -
              UPGRADE     )
        LISTCAT ENT(example.iam.aix) ALL
/*
```

Figure 35: Example of defining an IAM Alternate Index (EX1061A)

10.61 CONTINUED . . .

DEFINING AN IAM ALTERNATE INDEX WITH IAM ISPF PANELS

IAM ISPF PANEL Customers that have installed the IAM ISPF panels are able to define the IAM Alternate Index datasets using the IAM ISPF panels. The panels feature a fill in the blanks mechanism, along with being able to provide various relevant IAM overrides. After displaying the IAM Primary Option Menu, select option I for defining an IAM dataset. Be sure to fill in the dataset name of your alternate index using standard TSO naming conventions. Indicate that you are defining an alternate index by placing an X in the dataset type field. Optionally, you can specify an existing IAM or VSAM alternate index as a model for the dataset attributes. Below is an example of the IAM Primary Option Menu with parameters set for defining an IAM alternate index.

```
-------------------------- IAM PRIMARY OPTION MENU --------------------------
OPTION ===> i

   I     - Allocate (DEFINE) a new IAM Dataset                     Ver 8.0/01P
   V     - Allocate (DEFINE) a new VSAM Cluster
   D     - Delete a Dataset, Cluster, Path, or Alternate Index
   C     - Copy an IAM Dataset, VSAM KSDS, or VSAM ESDS
   M     - Move an IAM Dataset, VSAM KSDS, or VSAM ESDS
   R     - Rename a Dataset, Cluster, Path, or Alternate Index
   U     - Invoke an IAM Utility program
   blank - Display IAM Dataset or VSAM Cluster Information


Enter dataset name (required for all options except U)
 Dataset Name       ===> myiam.aix
 Dataset Type       ===> x  C=Cluster X=AIX P=Path (options I, V Only)

Enter Model or New dsname (optional for options I and V, required for option R)
 Model|Newname      ===>


Delete Confirmation ===> YES  Yes|No
```

Figure 36: IAM Primary Options Panel set for AIX Definition

10.61 CONTINUED . . .

IAM ALTERNATE INDEX DEFINITION ISPF PANEL

When you hit enter, the IAM Alternate Index Definition panel is displayed. If you had specified a MODEL dataset, the information from the model is filled in on the definition panel. You can set or change any of the attributes by filling in the blanks. Note that the fully qualified dataset name of the alternate index is displayed at the top left corner. Just below the alternate index name is a place for you to fill in the fully qualified name of the base cluster for this alternate index.

```
---------------------- DEFINE AN IAM ALTERNATE INDEX -----------------------
 COMMAND ===>

 IAM Alternate Index DSN: RAM2.MYIAM.AIX
 Related IAM Cluster DSN: ram2.myiam.cluster
                                        Multi-Volume Allocation ===> NO
 ALLOCATION
  Volume            ===> JUNK01        IAM OVERRIDES
  SMS Data Class    ===>                ANYVOL Unit         ===>
  Cyls|Recs|Trks    ===> cyls           Blocking Factor     ===>           1-15,>300
  Primary Space     ===> 2              Var. Overflow       ===> YES       Yes|No
  Secondary Space   ===> 1              Space Release       ===>           Yes|No
  Recatalog         ===> NO             Data Compress       ===>           Yes|No
                                        Minbufno            ===>           1-255
 ATTRIBUTES                             Maxbufno            ===>           1-255
  Max Recordsize    ===> 49
  Avg Recordsize    ===> 13             Upgrade             ===> yes       Yes|No
  Key Length        ===> 4              Unique Keys         ===> no        Yes|No
  Key Offset        ===> 16
  CI Size           ===> 4096          RETENTION
  CI/CA Free %      ===> 10  / 5        DAYS                ===>           0-9999
  Shareoption       ===> 2              EXPIRATION DATE     ===>           YYYY.DDD
```

Figure 37: Define an IAM Alternate Index ISPF Panel

In the above example, the information displayed in bold print represents what was provided by the user. Once all of the information is complete, you can hit enter, and IAM will allocate the alternate index. When completed, IAM will automatically return to the IAM Primary Option Panel, with the message DATASET ALLOCATED in the upper right hand corner. By hitting enter again, you can display the newly defined alternate index's attributes.

10.61 CONTINUED . . .

IAM ALTERNATE INDEX FILE CHARACTERISTICS DISPLAY

IAM responds with the IAM Alternate Index File Characteristics display, as shown below. The various attributes, where applicable, are filled in. At the bottom left is a section labeled Associations. This will contain related base cluster and the alternate index dataset itself. Once one or more paths have been defined for this alternate index, they will also be included under the associations.

```
---------------- IAM Alternate Index File Characteristics -----------------
COMMAND ===>
Dataset Name: RAM2.MYIAM.AIX

Definition Information              Allocation Information
   Record Length:          49          Volume:              JUNK01
   Record Format:    VARIABLE          Device Type:           3380
   Key Length:              4          Tracks in use:            1
   Key Offset:              5          Block Size:           11476
   Key Offset(BASE):       16          Blocking Factor:          4
   Unique Keys:           YES          Alloc Type:       CYLINDERS
   Share Option:            2          Primary Alloc:            1
   Release:                NO          Secondary Alloc:          1
   Storage Class:
   Data Class:
   Mgmt Class:

Statistics                          Extended Area Information
   Creation:         2002.315          Overflow Records:         0
   Expiration:       0000.000          Blocks Allocated:         0
   Last Reference:   2002.315          Blocks Used:              0
   Records:               256          Overflow Blocks:          0
   Deletes:                 0          PE Blocks:                0
   Inserts:                 0          Blocks Available:         0
   Updates:                 0          Variable Overflow:      YES
   Minimum Buffers:
   Maximum Buffers:

Associations
   CLUSTER-RAM2.MYIAM.CLUSTER
       AIX-RAM2.MYIAM.AIX
      PATH-RAM2.MYIAM.PATH
```

Figure 38: IAM Alternate Index File Characteristics Panel

10.62 BUILDING ON IAM ALTERNATE INDEX

BUILDING AN IAM ALTERNATE INDEX The process of building an IAM alternate index is identical to the VSAM process. The alternate index is usually built by use of the IDCAMS BLDINDEX command. However, other programs that have been written to build alternate indexes for VSAM can be used to build them for IAM. The internal record format of the IAM alternate index record is identical to the VSAM format.

USING IDCAMS BLDINDEX The BLDINDEX command for IDCAMS has two main operands, specifying the input file, or dataset, which is the base cluster, and specifying the output file or dataset, which is the alternate index. As long as there is sufficient virtual storage to sort the alternate key / prime key pairs, or if you have DFSORT, the example shown below of using IDCAMS BLDINDEX can be used. For complete information on using the IDCAMS BLDINDEX please refer to the IBM manual "DFSMS/MVS ACCESS METHOD SERVICES for the Integrated Catalog Facility", number SC26-4906.

```
//BLDINDEX   EXEC   PGM=IDCAMS
//SYSPRINT    DD    SYSOUT=*
//IAMINFO     DD    SYSOUT=*
//BASE        DD    DISP=OLD,DSN=example.iam.cluster
//AIX         DD    DISP=OLD,DSN=example.iam.aix
//SYSIN       DD    *
       BLDINDEX    INFILE(BASE)   OUTFILE(AIX)
/*
```

Figure 39: Example of IDCAMS BLDINDEX (EX1062A)

There are a few things to notice on the above example. First, the base cluster is allocated with a disposition of OLD. This is done to prevent updates from occurring while the alternate index is being built that could cause an out of synch condition with the alternate index. Second, there is an optional IAMINFO DD card in the JCL. This will cause IAM to produce an activity report for each IAM dataset that has been used.

If you do not have DFSORT and do not have enough virtual storage available for an internal sort, then you will need to add two work files for the BLDINDEX to work. The space requirements will depend on the size of your files, and up to five volumes can be specified for each work file. IDCAMS will define two ESDS type of files to these DD statements to use for sorting the data. The default DD names used by IDCAMS are IDCUT1 and IDCUT2. An example is shown below of using the work files.

```
//BLDINDEX   EXEC   PGM=IDCAMS
//SYSPRINT    DD    SYSOUT=*
//IAMINFO     DD    SYSOUT=*
//IDCUT1      DD    DSN=workfile1,VOL=SER=volser,UNIT=unitname,
//                  DISP=OLD,AMP='AMORG'
//IDCUT2      DD    DSN=workfile2,VOL=SER=volser,UNIT=unitname,
//                  DISP=OLD,AMP='AMORG'
//BASE        DD    DISP=OLD,DSN=example.iam.cluster
//AIX         DD    DISP=OLD,DSN=example.iam.aix
//SYSIN       DD    *
       BLDINDEX    INFILE(BASE)    OUTFILE(AIX)
/*
```

Figure 40: Example of IDCAMS BLDINDEX with work files (EX1062B)

10.62 CONTINUED . . .

USING IDCAMS BLDINDEX (Continued)

Using IAM ISPF Panels to Build an Alternate IndexThe IAM ISPF panels offer a build index function as well, that will invoke the IDCAMS BLDINDEX function. To go to the build index function from the IAM Primary Options panel select option U for the utilities panel. From the IAM Utility Program Menu, as shown below, select option I to build the alternate index.

```
--------------------- IAM UTILITY PROGRAM SELECTION MENU --------------------
 OPTION ===> i

    S  Specify utility print dataset allocation parameters

    I  BLDINDEX - Build Alternate Index
    R  IAMRECVR - IAM file diagnostic and recovery utility
    V  IAMSTATS - IAM VSAM Interface (VIF) module information
    X  IAMXMONI - IAM Execution Monitor/ISPF for CICS Regions
    Z  IAMZAPOP - IAM options table utility
```

Figure 41: IAM ISPF Utility Selection Menu

After hitting enter, IAM will display the Build Index panel. On this panel, enter the name of the base cluster and the name of the alternate index you want to build. After entering that information, press the enter key.

```
---------------------------- BLDINDEX - Datasets ----------------------------
COMMAND ===>

   Base Cluster ===> myiam.cluster

Alternate Index ===> myiam.aix

Specify BASE and AIX dataset names and hit ENTER
```

Figure 42: Example of IAM ISPF Build Index Panel

When the Build Index function is complete, IAM ISPF will open a browse panel of the output from the IDCAMS BLDINDEX function. An example is shown below:

```
Menu  Utilities  Compilers  Help
BROWSE    SYS02312.T164235.RA000.RAM2.R0100091            Line 00000000 Col 001 120
Command ===>                                                        Scroll ===> CSR
****************************************** Top of Data
*****************************************
IDCAMS  SYSTEM SERVICES                      TIME: 16:42:37        11/08/02     PAGE
1
BLDINDEX INFILE(DISKIN) OUTFILE(DISKOUT)
IDC0652I RAM2.MYIAM.AIX SUCCESSFULLY BUILT
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
IDC0002I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
**************************************** Bottom of Data
***************************************
```

Figure 43: Sample Output from BLDINDEX Function

10.63 DEFINING ON IAM PATH

DEFINING A PATH The process of defining a PATH to an IAM alternate index or base cluster is essentially the same as with VSAM. IAM will automatically determine if the PATHENTRY is an IAM dataset, so use of the OWNER($IAM) parameter is optional. An IAM path is a one-track dataset. The allocation request will be made for the same volume on which the path entry dataset resides. If the path entry dataset is DFSMS managed, the path dataset will be allocated with the same SMS classes as the path entry dataset. IAM uses the one track dataset to store information about the relationship of the path to it's alternate index or base cluster, rather than using a catalog entry. The IAM path dataset does not have to be on the same volume with the related dataset, although IAM attempts to place it there.

IDCAMS DEFINE PATH IDCAMS can be used to define a path. On the DEFINE PATH command, the PATHENTRY keyword indicates the related dataset. To access a base cluster through an alternate index, the PATHENTRY will be the name of the alternate index. The attribute UPDATE (or NOUPDATE) indicates whether the upgrade set of the alternate indexes will also be opened. The upgrade set includes all alternate indexes defined with the UPGRADE attribute.

If only the base cluster is being opened for update, any upgradeable alternate index datasets will also be opened. To prevent the alternate index upgrades from being done, define a PATH for the base cluster with the NOUPDATE attribute.

An example of defining an IAM PATH with IDCAMS is shown below.

```
//DEFPATH    EXEC   PGM=IDCAMS
//SYSPRINT    DD    SYSOUT=*
//SYSIN       DD    *
       DEFINE PATH                  -
       (NAME(example.iam.path)      -
       PATHENTRY(example.iam.aix)   -
       UPDATE )
/*
```

Figure 44: Example of Using IDCAMS to Define a Path (EX1063A)

10.63 CONTINUED . . .

IAM ISPF DEFINE PATH

If you have the IAM ISPF panels installed, they also provide a Define Path function. Starting at the IAM Primary Options menu, as shown below, select option I to define an IAM dataset, then enter the name of the path that you want to define, and indicate a P for dataset type, indicating a path. An example of the completed panel is shown below.

```
--------------------------- IAM PRIMARY OPTION MENU ---------------------------
OPTION ===> i

   I      - Allocate (DEFINE) a new IAM Dataset                     Ver 8.0/01P
   V      - Allocate (DEFINE) a new VSAM Cluster
   D      - Delete a Dataset, Cluster, Path, or Alternate Index
   C      - Copy an IAM Dataset, VSAM KSDS, or VSAM ESDS
   M      - Move an IAM Dataset, VSAM KSDS, or VSAM ESDS
   R      - Rename a Dataset, Cluster, Path, or Alternate Index
   U      - Invoke an IAM Utility program
   blank - Display IAM Dataset or VSAM Cluster Information


Enter dataset name (required for all options except U)
 Dataset Name        ===> MYIAM.PATH
 Dataset Type        ===> P  C=Cluster X=AIX P=Path (options I, V Only)

Enter Model or New dsname (optional for options I and V, required for option R)
 Model|Newname       ===>


Delete Confirmation ===> YES  Yes|No
```

Figure 45: IAM ISPF Define of a Path

After hitting enter, IAM will display the IAM Define Path panel as shown below. On this panel, enter the fully qualified name of the path entry dataset, which is either an alternate index or a base cluster. Also enter whether or not the upgrade set is to be opened if the path is opened for update. While a specification of NO will eliminate the overhead of updating all of the alternate indexes in the upgrade set, the alternate indexes will not properly reflect the contents of the base cluster.

```
----------------------------- DEFINE AN IAM PATH -----------------------------
COMMAND ===>

 Path Name ===> RAM2.MYIAM.PATH
Path Entry ===> ram2.myiam.aix


    Update ===> YES  Yes = Open all UPGRADE sets when PATH opens
                      No = Open only BASE CLUSTER when PATH opens
```

Figure 46: IAM ISPF Define an IAM Path Panel

After hitting enter, IAM will return to the IAM Primary Options panel, with a message in the upper right hand corner indicating that the path was successfully allocated. To display the attributes of the Path, hit enter again, with the Path name in the dataset name field.

10.63 CONTINUED . . .

IAM PATH CHARACTERISTICS

The IAM Path Characteristics panel displays basic information about the actual data record that would be retrieved from the base cluster through this path. Because the IAM Path dataset is an actual one track dataset, volser and device type information is displayed. At the bottom is the associations indicating the datasets related to this path.

```
------------------------ IAM Path Characteristics ------------------------
COMMAND ===>
Dataset Name: RAM2.MYIAM.PATH

Definition Information            Allocation Information
   Base Record Length:   128         Volume:             JUNK01
   Base Key Offset:        8         Device Type:          3380
   AIX Key Length          4         Tracks in use:           1
   Update:               YES         Blocking Factor:      4096
                                     Alloc Type:        RECORDS

Statistics
   Creation:        2002.315
   Expiration:      0000.000
   Last Reference:  2002.315

Associations
   CLUSTER-RAM2.MYIAM.CLUSTER
       AIX-RAM2.MYIAM.AIX
      PATH-RAM2.MYIAM.PATH
```

Figure 47: IAM Path Characteristics Panel

10.64 ACCESSING DATA THROUGH AN IAM ALTERNATE INDEX

Now that you have a base cluster defined and loaded, an alternate index defined and built, and a path defined, you are ready to start accessing your data through the alternate index. Just as with VSAM, all you need to do is to allocate (dynamically or with a DD card) the path to the alternate index, open it and you are ready to go. When your program opens the path, IAM will dynamically allocate and open all of the necessary components of the alternate index sphere. For example, if you were to just be opening the path for read only access, IAM will dynamically allocate the alternate index and the base cluster datasets, open them, and set up the control blocks and routines necessary for your program to access the data. If the program opens up the path for update, then any associated upgradeable alternate indexes will also be dynamically allocated and opened. IAM will use the system generated DD names for the files that are dynamically allocated, which have the format SYSnnnnn. When you close the path, IAM will close and deallocate all of the component datasets that had been opened.

Because the DD names used to access the component IAM datasets in the alternate index sphere can not be known in advance, the IAM override processor has been enhanced to handle specification of overrides by dataset name (DSN=) in addition to the DD name. Using the DSN= parameter on an IAM ACCESS override is valid for any enhanced format IAM file. The DD= override can still be specified on other override cards, which can be intermixed with the new DSN= parameter. Information on the IAM ACCESS overrides is in section 10.40 or section 30.03. An example of using a path, along with IAM ACCESS overrides by DSN= is shown below:

```
//UPDATAIX     EXEC   PGM=mypgm
//PATH          DD    DISP=SHR,DSN=example.iam.path
//IAMINFO       DD    SYSOUT=*
//IAMOVRID      DD    *
       ACCESS         DSN=example.iam.base,MAXBUFNO=32
       ACCESS         DSN=example.iam.aix,DYNCORE=1024
/*
```

Figure 48: Example of JCL to use a Path with IAM Overrides

10.65 SPECIAL CONSIDERATIONS WITH IAM ALTERNATE INDEXES

RENAMING AND COPYING There are some considerations when any associated dataset, or all of the associated datasets, in an alternate index sphere are renamed. This same consideration exists if the datasets are copied or restored with a new name. Because IAM datasets appear to the utilities performing such functions as non-VSAM datasets, the association information is not automatically updated when a dataset name is changed. To update the association information, a DEFINE RECATALOG must be done on all of the affected alternate index and path datasets. First, the base cluster should be recataloged, followed by the alternate index dataset(s), and then the path datasets. The information that must be provided includes the new dataset name, the new related or path entry dataset name, and for base clusters and alternate index datasets, the volume(s). The base cluster RECATALOG must also include the OWNER($IAM) parameter, which is optional for the Alternate Index and Path recatalog. The IAM alternate index support intercept for the DEFINE RECATALOG function has processing to reset the association names. An example of a DEFINE RECATALOG is shown below.

```
//RECAT      EXEC   PGM=IDCAMS
//SYSPRINT    DD    SYSOUT=*
//SYSIN       DD    *
      DEFINE CLUSTER                                -
             (NAME(newname.iam.cluster)             -
             OWNER($IAM)                            -
             VOLUME(volser)                         -
             RECATALOG    )
      DEFINE ALTERNATEINDEX                         -
             (NAME(newname.iam.aix)                 -
             RELATE(newname.iam.cluster)            -
             VOLUMES(volser)                        -
             RECATALOG    )
      DEFINE PATH                                   -
             (NAME(newname.iam.path)                -
             PATHENTRY(newname.iam.aix)             -
             RECATALOG    )
      LISTCAT ENT(newname.iam.cluster)   ALL
      LISTCAT ENT(newname.iam.aix)       ALL
      LISTCAT ENT(newname.iam.path)      ALL
/*
```

Figure 49: Example of Recatalog after a Rename (EX1065A)

REMOVING ASSOCIATION INFORMATION If you have only made a copy of the base cluster and renamed it, you can eliminate the association information from the base cluster by issuing a DELETE clustername AIX NOSCRATCH. IAM will recognize this special delete and remove all of the associations from its definition. This delete must be done prior to performing a recatalog as described above, because IAM checks before erasing the association information that the current base cluster name does not match the base cluster name in the associations.

DELETE SUPPORT IAM alternate index support includes special processing for dataset deletion. As with VSAM, when the base cluster is deleted, all related alternate index and path datasets will also be deleted, providing that the SVC 26 DELETE interface is used. This interface also requires that the related datasets be cataloged. If a related dataset is not cataloged, it will be skipped over by the delete process. This is the interface that is used by IDCAMS, the TSO DELETE command, and the IAM ISPF panels. If the SCRATCH SVC interface is used for non-SMS managed volumes, then IAM will not intercept the request, and only the specified dataset will be deleted.

If a delete is issued for an alternate index, then the alternate index and any related paths will be deleted. If the delete is issued for a path, only the path will be deleted. These actions are consistent with VSAM delete processing.

10.70 IAM REPORTS

REPORTING OVERVIEW

IAM offers a variety of reports to assist in the management of IAM datasets. The two primary reports are the IAMPRINT report and the IAMINFO report. The IAMPRINT report is automatically produced when a LISTCAT ALL is done on an IAM dataset. The IAMINFO report is a run time report that is produced whenever an IAM dataset is closed, providing that there is an IAMINFO DD card allocated to the job. The IAMINFO reports can also be generated from the IAM SMF records, if they are collected, using the IAMSMF program. The IAMSMFVS program produces a more compact set of reports, based on an accumulation of the IAM SMF records for each dataset. There are three different reports produced by IAMSMFVS, all of which have one line per dataset.

10.71 IAMPRINT REPORT

LISTCAT REPORT

The IAMPRINT report is produced by an IDCAMS LISTCAT ALL command. This report contains information about the dataset attributes, the DASD volumes being used, along with the amount of space on each volume, plus various statistics about the dataset. The statistics that are presented in the IAMPRINT report are generally only updated when the dataset is closed. If the dataset is currently open for update the statistics may not accurately reflect the current status of the dataset. A message will appear at the bottom of the report if this is determined to be true by the IAM LISTCAT processor. The statistics can also be invalid due to a system failure that prevented the dataset from being properly closed, or if the dataset was being concurrently shared by multiple jobs for update processing, without going through IAM RLS. **IAM does not support or provide for data integrity when a dataset is shared for concurrent update purposes unless IAM RLS or other software that enables sharing is used. Doing so may result in the loss of data.**

LISTCAT EXAMPLE

Specification of the IAMPRINT DD is optional. If the IAMPRINT DD is not specified, then IAM will allocate one to SYSOUT=* for batch jobs, providing that the SYSPRINT DD is also allocated to a SYSOUT class. For TSO users, the IAM output will be displayed directly on the screen. In both cases, the report will be produced on the IAMPRINT DD dataset, if the file was explicitly specified by the user. The IAM LISTCAT processing can be disabled by placing an //IAMNOLIC DD DUMMY card in the job step performing the LISTCAT(s). Using the IAMNOLIC DD card is intended for jobs that are performing generic LISTCAT operations, and the IAM information is not necessary.

```
//LISTCAT   EXEC PGM=IDCAMS
//SYSPRINT DD    SYSOUT=*
//SYSIN     DD    *
  LISTCAT ENT(IAMV.VAR100.CLUSTER) ALL
/*
```

Figure 50: JCL to Obtain IAM Listcat (IAMPRINT) Output (EX1071A)

LISTCAT OUTPUT

The results of running the above job on an IAM dataset are two printed reports. The first report is produced by IDCAMS on SYSPRINT that displays the information on the actual non-VSAM catalog entry for the IAM dataset. The second report is produced by IAM on IAMPRINT that displays the attributes and other information about the dataset itself. The two reports are shown below.

```
IDCAMS  SYSTEM SERVICES       TIME: 10:34:29    06/18/02    PAGE

  LISTCAT ENTRIES(IAMV.VAR100.CLUSTER) ALL
NONVSAM ------- IAMV.VAR100.CLUSTER
      IN-CAT --- IAMVT.CATALOG
      HISTORY
        DATASET-OWNER-------$IAM    CREATION--------2002.169
        RELEASE----------------2    EXPIRATION------0000.000
      VOLUMES
        VOLSER------------SCR092    DEVTYPE------X'3010200F'    FSEQN------------------0
  ASSOCIATIONS--------(NULL)
  ATTRIBUTES
```

Figure 51: IDCAMS LISTCAT Output from SYSPRINT

As you can see, the IDCAMS output indicates that the dataset type is non-VSAM. While the OWNER field is set to '$IAM', it is not a reliable indicator that the dataset is in fact an IAM dataset.

10.71 CONTINUED . . .

IAMPRINT OUTPUT Below is an example of the IAMPRINT Report for an Enhanced Format IAM dataset. The fields displayed on the report will vary, depending on the type and status of the IAM file.

```
IAM400 IAM CATALOG INFORMATION SERVICE -- VER 8.0/01P  DATE 2002.169 TIME 10:34:30
IAM100 IAM FILE ANALYSIS - DSN=IAMV.VAR100.CLUSTER
       ------------------------------------------------------------------------
       FILE FORMAT -- =  ENHANCED  -  FILE STATUS ---------- =    LOADED        -
       RECORD SIZE -- =      2040  -  FREESPACE - CI% ------- =        10        -
       CI SIZE ------ =      2048  -  FREESPACE - CA% ------- =        10        -
       BLOCK SIZE --- =     13682  -  EXTENDED OVERFLOW ----- =      7901 RECS   -
       BLOCK FACTOR - =         4  -  REQUESTED OVERFLOW ---- =     50000 RECS   -
       VAR OVERFLOW - =       YES  -  EXTENDED OVERFLOW ----- =      1317 BLOCKS -
       KEY SIZE ----- =        58  -  EXTENDED OVERFLOW USED  =        16 %      -
       KEY OFFSET --- =         8  -  EXTENDED PE ---------- =         0 BLOCKS -
       FILE TYPE ---- =      KSDS  -  EXTENDED ALLOCATED ---- =      8348 BLOCKS -
       DEVICE TYPE -- =      3390  -  EXTENDED AVAILABLE ---- =      6992 BLOCKS -
       VOLUME COUNT - =         1  -  SPACE USED ----------- =      1387 TRACKS -
       VOLSER ------- =    SCR092  -  SPACE ALLOCATED ------- =      3135 TRACKS -
       TOTAL EXTENTS  =         1  -  TOTAL SPACE ALLOCATED - =      3135 TRACKS -
       PRIMARY SPACE  =       500  -  SECONDARY SPACE ------- =       100 CYL    -
       MULTIVOLUME -- =   PRIMARY  -  MAX SECONDARY --------- =       500 CYL    -
       RELEASE ------ =        NO  -  SHARE OPTIONS --------- =         2        -
       DATA COMPRESS  =   ENABLED  -  INDEX COMPRESS -------- =       YES        -
       TOTAL RECORDS  =     46000  -  INSERTS --------------- =     10000        -
       UPDATES ------ =         0  -  DELETES --------------- =         0        -
       HIGH USED RBA  =  75907736  -  HIGH ALLOCATED RBA ---- = 171572280        -
       FILE DEFINED - =  2002.169 -  06/18/2002 - 10:31 AM - =  10:31:24         -
       FILE LOADED -- =  2002.169 -  06/18/2002 - 10:31 AM - =  10:31:46         -
       LAST UPDATED - =  2002.169 -  06/18/2002 - 10:33 AM - =  10:33:42         -
       STORAGE REQUIRED FOR COMPRESSED PRIME INDEX -- =    138646                -
       NUMBER OF IAM DATA BLOCKS ------------------ =      4163                  -
       EXTENDED HIGH ALLOCATED RBN ---------------- =     12539                  -
        ------------------------------------------------------------------------

IAM499 IAMLISTC(8.0/01P ) PROCESSING COMPLETED
```

Figure 52: Example of IAMPRINT Output of an Enhanced Format Dataset

10.71 CONTINUED . . .

IAMPRINT OUTPUT FOR AN ALTERNATE INDEX

The IAMPRINT output for an alternate index is almost identical to the report for a base cluster. An IAM Alternate Index is essentially an IAM KSDS type of file. The file type indicates that this is an AIX (Alternate Index) dataset. Just below the DATA COMPRESS line is an additional line, that includes the alternate key offset in the base record, and to the right of that is the indication of whether or not the alternate index keys are unique. At the bottom of the report is the associations. The associations will also appear on a base cluster once there are defined alternate indexes and paths to that base cluster.

```
IAM400 IAM CATALOG INFORMATION SERVICE -- VER 8.0/01P  DATE 2002.169 TIME 09:47:23
IAM100 IAM FILE ANALYSIS - DSN=IAMV.AIX112.AIX1
       ---------------------------------------------------------------------------
       FILE FORMAT -- =  ENHANCED  -  FILE STATUS ----------- =     LOADED        -
       RECORD SIZE -- =       733  -  FREESPACE - CI% ------- =         10        -
       CI SIZE ------ =      1024  -  FREESPACE - CA% ------- =         10        -
       BLOCK SIZE --- =     11476  -  EXTENDED OVERFLOW ----- =          0 RECS   -
       BLOCK FACTOR - =         4  -  REQUESTED OVERFLOW ---- =          0 RECS   -
       VAR OVERFLOW - =       YES  -  EXTENDED OVERFLOW ----- =          0 BLOCKS -
       KEY SIZE ----- =         4  -  EXTENDED OVERFLOW USED  =          0 %      -
       KEY OFFSET --- =         5  -  EXTENDED PE ----------- =          0 BLOCKS -
       FILE TYPE ---- =       AIX  -  EXTENDED ALLOCATED ---- =          0 BLOCKS -
       DEVICE TYPE -- =      3380  -  EXTENDED AVAILABLE ---- =          0 BLOCKS -
       VOLUME COUNT - =         1  -  SPACE USED ------------ =          5 TRACKS -
       VOLSER ------- =    SCR081  -  SPACE ALLOCATED ------- =         15 TRACKS -
       TOTAL EXTENTS  =         1  -  TOTAL SPACE ALLOCATED - =         15 TRACKS -
       PRIMARY SPACE  =         5  -  SECONDARY SPACE ------- =          1 CYL    -
       MULTIVOLUME -- =   PRIMARY  -  MAX SECONDARY --------- =          5 CYL    -
       RELEASE ------ =        NO  -  SHARE OPTIONS --------- =          1        -
       DATA COMPRESS  =   ENABLED  -  INDEX COMPRESS -------- =         NO        -
       AIX KEY OFFSET =        16  -  UNIQUE KEYS ----------- =         NO        -
       TOTAL RECORDS  =     10000  -  INSERTS --------------- =          0        -
       UPDATES ------ =         0  -  DELETES --------------- =          0        -
       HIGH USED RBA  =    229520  -  HIGH ALLOCATED RBA ---- =     229520        -
       FILE DEFINED - =  2002.169  -  06/18/2002 -  9:46 AM - =   09:46:41        -
       FILE LOADED -- =  2002.169  -  06/18/2002 -  9:46 AM - =   09:46:55        -
       STORAGE REQUIRED FOR PRIME INDEX ------------- =        68                 -
       NUMBER OF IAM DATA BLOCKS ------------------- =         19                 -
       EXTENDED HIGH ALLOCATED RBN ----------------- =          0                 -
       ---------------------------------------------------------------------------

       ASSOCIATIONS -
       CLUSTER ------ = IAMV.AIX112.CLUSTER
       AIX ---------- = IAMV.AIX112.AIX1                             UPGRADE
       PATH --------- = IAMV.AIX112.PATH1                            UPDATE
IAM499 IAMLISTC(8.0/01P ) PROCESSING COMPLETED
```

Figure 53: Example of IAMPRINT for an IAM Alternate Index

10.71 CONTINUED . . .

IAMPRINT REPORT ON A PATH

The report for an IAM Path dataset is similar in format to the above report, only much shorter. An IAM Path is a 1-track dataset, which is used to provide access to a base cluster through an alternate index, or to the base cluster directly. The information of most interest and concern in this report is the associations. The Path serves as a connector that is used by OPEN processing to determine what datasets need to be allocated and opened.

The file attributes displayed, including record size, alternate key size, and alternate key offset will all reflect the data record as retrieved through the path. The record size will always be the base cluster maximum record size. If the Path is for an alternate index, then the alternate key size and alternate key offset will be as defined for the alternate index. If the Path is for a base cluster, then the alternate key size and alternate key offset will be the base cluster's key size and offset.

An example of an IAMPRINT report on a PATH is shown below.

```
IAM400 IAM CATALOG INFORMATION SERVICE -- VER 8.0/01P  DATE 2002.169 TIME 09:47:23
IAM100 IAM FILE ANALYSIS - DSN=IAMV.AIX112.PATH1
       ---------------------------------------------------------------------------
       FILE TYPE ---- =      PATH  -  ALTERNATE KEY SIZE ---- =        4         -
       RECORD SIZE -- =       500  -  ALTERNATE KEY OFFSET -- =       16         -
       VOLUME COUNT - =         1  -  SPACE USED ----------- =       N/A TRACKS -
       VOLSER ------- =    SCR081  -  SPACE ALLOCATED ------- =         1 TRACKS -
       TOTAL EXTENTS  =         1  -  TOTAL SPACE ALLOCATED - =         1 TRACKS -
       FILE DEFINED - =  2002.169  -  06/18/2002 -  9:47 AM - =  09:47:06        -
       ---------------------------------------------------------------------------

       ASSOCIATIONS -
       CLUSTER ------ = IAMV.AIX112.CLUSTER
       AIX ---------- = IAMV.AIX112.AIX1                               UPGRADE
       PATH --------- = IAMV.AIX112.PATH1                              UPDATE
IAM499 IAMLISTC(8.0/01P ) PROCESSING COMPLETED
```

Figure 54: Example of an LISTCAT Output for an IAM Path

10.71 CONTINUED . . .

IAMPRINT FIELD DESCRIPTIONS

The fields on the IAMPRINT report for an Enhanced format file are described below. The table starts with the fields on the left side of the report, and are followed by all of the fields on the right side of the report. The last portion describes the fields that appear as a single column at the bottom of the report. Note that not all fields are present in all variations of the report.

The header of the report has an IAM400 message followed by the IAM100 message. The message numbers are not displayed when the output is directed at a TSO terminal. The IAM400 message indicates that the report is from a catalog information request (i.e., LISTCAT) along with the version and level of IAM producing the report. The date and time of the report is included in that message. The dataset name is indicated on the IAM100 message.

LEFT COLUMN FIELDS

IAMPRINT Field (Left Column)	Description of Field Contents:
File Format	Indicates the file format use to store the data. Possible values are: • **ENHANCED –** Indicates that the file is an enhanced format IAM file. • **SMS-EXT –** Indicate that the file is a DFSMS extended format and an IAM enhanced format file. • **COMPATIBLE –** Indicates that the file is a compatible format file, which can be used with the older non-VSAM interfaces to IAM.
Record Size	The maximum amount of data that can be contained in a single record. Value is from the RECORDSIZE parameter on the IDCAMS DEFINE statement.
CI Size	The specified Control Interval (CI) size from the DEFINE. If CI size was not specified on the Define, IAM calculates a valid VSAM CI size based on the maximum record size.
Block Size	The physical block size IAM is using for the file. A block is the amount of data transferred in an I/O operation, and that is stored in a single contiguous stream of data on the actual device
Block Factor	Indicates the number of blocks per track, or the user specified block size from an IAM CREATE Override.
Var Overflow	Indicates if IAM is using variable length overflow for this file. If the value is **YES**, IAM will put as many records as can fit within each overflow block. If the value is **NO**, IAM will only place in each overflow block the number of maximum length records that will fit in an overflow block.
Key Size	For KSDS type of files, indicates the defined length for the key of each data record.
Key Offset	Specifies the relative position of the key, as an offset from the beginning of the record, where a value of 0 indicates the first byte. For an AIX type of dataset, this is the key offset within the alternate index dataset itself, and will always be 5.
File Type	Indicates the file type. Possible values are: • **KSDS –** Keyed sequence dataset • **ESDS –** Entry sequence dataset. • **ESDS/P –** Entry sequence dataset with PSEUDORBA. • **ESDS/EA –** Entry sequence dataset with 8-byte RBA values. • **AIX –** An IAM Alternate Index dataset • **PATH –** An IAM Path dataset.

10.71 CONTINUED . . .

Device Type Indicates the type of DASD device architecture on which the dataset currently resides, for example a 3380 or a 3390.

Volume Count The number of volumes contained in the catalog entry for this file.

Volser This indicates the volume(s) to which the dataset is cataloged. This line is repeated for each volume that is in the catalog entry for the file, except for SMS candidate volumes.

Total Extents Indicates the total DASD extents allocated to the file. An extent is a contiguous area of space on the device being used by the dataset.

Primary Space Indicates the requested primary space quantity that was specified when the file was originally defined.

Multivolume Has values of **PRIMARY** or **SECONDARY**, which indicates from which space parameter the size of the first extent on each DASD volume is allocated.

Release Indicates if DASD space will be released when the file is loaded again. Generally this is **YES** before a file has been loaded, and is set to NO after the first load.

Data Compress Indicates whether this file can contain IAM data compressed records. Possible values are:

- **YES –** Indicates file is eligible for IAM software compression.
- **HARDWARE –** Indicates file is eligible for IAM use of the IBM hardware compression function.
- **NO –** No data compression will be performed for this file.

Dictionary Indicates the name of the hardware compression dictionary that the user has specified for this file. This is only present when the file is enabled for hardware compression, and is using a customized dictionary.

AIX Key Offset For an Alternate Index type of dataset, indicates the offset of the alternate key within the base cluster.

Total Records The number of user data records in the file, as of the last close.

Updates The number of user data records that have been updated since the file was last loaded.

High Used RBA For IAM KSDS files, indicates the amount of space used for the file, in bytes. For IAM ESDS files, indicates the amount of actual user data contained within the file.

IAM Journal Indicates that IAM journaling has been specified, either through the JRNAD= IAM override, or through the LOG() IDCAMS define keyword. Possible values are **ENABLED** indicating that IAM journaling is active, or **NONE** indicating journaling has been explicitly disabled for this file.

10.71 CONTINUED . . .

RIGHT COLUMN FIELDS

IAMPRINT Field (Right Column)	Description of Field Contents:
File Status	Indicates if the file is **LOADED** or **UNLOADED**. A file will be in the UNLOADED state if it has been defined but has not had a successful load, or if a file load or reorganization has failed or is in progress.
Freespace - CI%	The amount of space to be left free in each block, as a percentage of the block size, as the file is being loaded or extended. Specified on the file definition. For IAM files, this is referred to as the Integrated Overflow Area.
Freespace - CA%	The amount of CA% freespace specified on the file definition. This is used by IAM to calculate an amount of allocated but unused space that is to be retained by the file after the initial load.
Extended Overflow - Recs	The number of records in the Extended Overflow area of the file, as of the last time the file was closed.
Requested Overflow - Recs	This is the overridden value for the size of the overflow area, in records, when the file was defined or loaded. This value, if provided, will be used when calculating the percentage of overflow used.
Extended Overflow - Blocks	The number of blocks assigned to the Extended Overflow area.
Extended Overflow Used	Indicates, as a percentage, the amount of overflow space used. If a value for overflow records is provided as an override (O=), then the used percentage is based on that value. Otherwise, this is the percentage of the currently allocated extended blocks that are being used.
Extended PE	The number of Extended blocks assigned as Prime Extension (PE) blocks.
Extended Allocated	The total number of extended area blocks that the file can contain within the currently allocated extents.
Extended Available	The number of extended area blocks that are available for use, which could be assigned to either Extended Overflow, or Extended PE.
Space Used	The amount of DASD space currently required for the file. (Does not include space required for Extended Available blocks.)
Space Allocated	For each volume, indicates the amount of DASD space allocated, in tracks.
Total Space Allocated	Total DASD space allocated, for all volumes.
Secondary Space	Indicates the amount of space to be requested when a secondary extent is required, as specified on the file define.
Max Secondary	The maximum amount of DASD space IAM will request, when additional DASD space is needed for this dataset.
Share Options	Indicates the defined cross-region share option.
Index Compress	Indicates whether a compressed index structure exists for this file.

10.71 CONTINUED . . .

Dictionary Saved When the IAM file is compressed by hardware, indicates either YES that the dictionary has been saved within the dataset, or NO that the dictionary is not contained within the dataset.

Unique Keys For an alternate index, indicates if the alternate keys are UNIQUE (YES) or NONUNIQUE(NO). Unique keys means that each alternate key can only index one data record in the base cluster. A value of NO means non-unique keys, where each alternate key can index multiple data records in the base cluster.

Inserts Indicates the number of records added since the file was last loaded.

Deletes Indicates the number of records deleted since the file was last loaded.

High Allocated RBA Total number of bytes of DASD storage allocated to the file. For ESDS type files, this value can actually be lower than the High Used RBA field. The reason is that the High Used is based on the length of the actual user data. If the file is data compressed, the high allocated could be substantially lower.

Alternate Key Size Field is printed for a Path type of dataset only, indicates the size of the key for the data accessed via this path. For a path on an alternate index, this will be the alternate key size. For a path on a base cluster, this will be the primary key size.

Alternate Key Offset Field is printed for a Path type of dataset only. This is the key offset for the data as accessed via this path. For a path on an alternate index, this is the offset of the alternate key within the prime record. For a path on a base cluster, this is the offset of the primary key.

Journal Records Indicates what type of journal records will be produced. Possible values include ALL, BEFORE, AFTER or NONE.

10.71 CONTINUED . . .

CENTER COLUMN FIELDS

IAMPRINT Field (Center Column)	Description of Field Contents:
File Defined	The date and time when the file was last defined. The date can be in either mm/dd/yyyy format, or if the EURODATE Global Option is set, will be in the dd/mm/yyyy format.
File Loaded	The date and time that the last file load was completed. However, if a file load had started, but is not yet complete, this time will be the starting time of the last attempted file load.
Last Updated	The date and time of the last close of the file from a program that updated the file.
Pseudo Maximum Logical Record Length	If a PSEUDOLRECL value was specified when the file was defined, this field will appear with the value that was specified. Programs doing a SHOWCB for the LRECL field, or a SHOWCAT will be returned this value for the maximum record length in the dataset.
Storage Required for (Compressed) Prime Index	Indicates the amount of virtual storage that is required to contain the index to the Prime Data area of the file, and whether or not the index has been compressed.
Number of IAM Data Blocks	Indicates the number of blocks in the dataset preceding the index area. This value may need to be supplied via the MAXBLKS parameter to IAMRECVR on a RECOVER operation.
Extended High Allocated RBN	Indicates the high allocated block number, to be used when it is necessary to run IAMRECVR to recover a file after it has been damaged. Value is specified via the XTNDEDHARBN keyword.

There are a few additional informational messages that may appear based on the status of the file at the time the LISTCAT was done. Unless they do not reflect the actual file status, they generally are of no major concern. These messages include:

- FILE HAS NOREUSE ATTRIBUTE, CAN NOT BE RELOADED
- FILE REORG HAS STARTED, AND HAS NOT COMPLETED
- FILE REORGANIZATION IS RECOMMENDED
- FILE IS OPEN, STATISTICS MAY BE INACCURATE

ASSOCIATIONS For IAM datasets that are part of an Alternate Index Sphere, there will be an Associations section of the report. This section will follow all of the above described sections, and include an entry for each associated dataset. The dataset name will be preceded by the type of dataset (Cluster, AIX, or Path). Then after each AIX entry name, the UPGRADE or NOUPGRADE attribute will be displayed. For Path's, the UPDATE or NOUPDATE attribute will be displayed. The dataset names displayed are those that have been defined. If one or more of the associated datasets has been renamed, copied to a new name, or restored with a new name, then a DEFINE RECATALOG must be performed to update the association information. See Section 10.65 for information on performing a Define Recatalog.

10.72 IAMINFO REPORT

IAMINFO REPORTS

IAM will provide a one page run time report on dataset usage, which is produced each time an IAM dataset is closed providing that the job has an IAMINFO DD card. IAM will also optionally produce an SMF record with the same information, if so enabled by the IAM Global Options table. These reports contain information on the dataset attributes, various statistics about the content of the dataset, statistics on the resource usage, and statistics on the different types of I/O requests processed. The IAM Overrides used for each dataset are also included in the report, adjacent to the statistic that would be most influenced by the override. The IAMINFO reports are the primary information source for what is going on with each dataset. Many of the questions about the resource usage or performance on any particular IAM dataset can be answered with the IAMINFO reports.

The IAMINFO reports are easily obtained by providing an IAMINFO DD card in each job step that processes an IAM dataset. Normally this DD statement indicates a SYSOUT file, but can indicate an actual sequential dataset if desired. The only additional overhead when using IAMINFO is the overhead of actually formatting and writing out the report. IAM always maintains the various statistics that are reported on with IAMINFO, regardless of whether or not the DD statement is in the JCL.

IAM SMF RECORDS

IAM can also optionally produce SMF records that contain all the information that appears on the IAMINFO reports. To enable this facility, the IAM Global Option SMF=YES must be specified, along with indicating the SMF record type to use by specifying RECTYPE=nnn. IAM provides the IAMSMF program that can produce IAMINFO reports from the SMF data. There is also the IAMSMFVS program, which produces more of a summarization set of reports from the IAM SMF data. Additionally, customers can provide their own programs to report on the IAM SMF data.

EXAMPLE OF AN IAMINFO DD CARD

As can be seen from the example below, all that needs to be done to obtain IAMINFO reports for job steps that process IAM datasets is to add an IAMINFO DD card.

```
      //PROCESS   EXEC PGM=anypgm
      //SYSPRINT   DD    SYSOUT=*
      //iamfile    DD    DISP=SHR,DSN=my.iam.dataset
  →   //IAMINFO    DD    SYSOUT=*                        ← added DD card
```

Figure 55: Example of adding an IAMINFO DD card (EX1072A)

10.72 CONTINUED . . .

IAMINFO FOR FILE LOAD

There are two basic IAMINFO reports. The first is for a file load, for which a sample is provided below. The file load is indicated in the IAM361 message, where it indicates that the dataset was opened for FILE CREATION. Amongst the unique features of the IAMINFO file load report are that the average and maximum record lengths are reported on. These lengths are recorded after data compression, if any. Also reported on, under the IAM365 message, is the amount of data space used as a temporary work area for the index. If the DISK BLOCKS READ is not zero, then a temporary dataset was used for the work area for the index, instead of a Data Space.

```
IAM400 INNOVATION IAM INFORMATION PRINT ROUTINE--IAMNINFO VER 8.0/01P--INNOVATION DATA PROCESSING DATE-2002.169

 IAM360  STEP - VAR100D    DDNAME - VSAMCRT1   DATASET MONITORED - IAMV.VAR100.CLUSTER
 IAM361  INFO REQUESTED BY PROGRAM IAMTVSAM FOR FILE CREATION      OPENED-2002.169.13:47:32  CLOSED-2002.169.13:47:53
 IAM362  IAM DATA CHARACTERISTICS -                                                             OVERRIDES IN EFFECT:
         IAM FILE FORMAT-----------=   ENHANCED    -    IAM FILE STATUS-----------=     LOADED
         LOGICAL RECORD LENGTH-----=       2040    -    CI SIZE-------------------=       2048
         KEY SIZE------------------=         58    -    KEY OFFSET----------------=          8
         AVERAGE RECORD LENGTH-----=       1338    -    LARGEST RECORD PROCESSED--=       2000

 IAM363  IAM FILE CHARACTERISTICS -
         BLOCKING FACTOR-----------=          4    -    BLOCK SIZE----------------=      13682
         TRACKS IN USE-------------=       1048    -    VARIABLE LENGTH OVERFLOW--=        YES
         DATASET TYPE--------------=       KSDS    -    SHARE OPTIONS-------------=          2
         NUMBER OF IAM DATA BLOCKS-=       4163    -    HIGH ALLOCATED RBN--------=          0   RELEASE=NO
         INTEGRATED OVERFLOW (CI%)-=         10    -    DASD RESERVE (CA%)--------=         10
         FILE DEFINED DATE---------=   2002.169    -   FILE DEFINED TIME---------=   13:47:30
         FILE LOADED DATE----------=   2002.169    -   FILE LOADED TIME----------=   13:47:53

 IAM372  IAM EXTENDED AREA CHARACTERISTICS -
         EXT. OVERFLOW RECORDS-----=          0    -    EXT. OVERFLOW BLOCKS------=          0
         EXTENDED BLOCKS ALLOCATED-=          0    -    EXTENDED PE BLOCKS--------=          0
         EXTENDED BLOCKS USED -----=          0    -    EXTENDED BLOCKS AVAILABLE-=          0

 IAM365  IAM EXECUTION STATISTICS -
         TOTAL STORAGE REQUIRED----=    1996024    -    PRIME INDEX(COMPRESSED)---=     138646
         STORAGE ABOVE THE LINE----=    1971192    -    COMPRESSED DATA STRUCTURE-=        YES
         REQUESTS PROCESSED--------=      36002    -    REQUESTS FAILED-----------=          0
         DISK BLOCKS READ----------=          0    -    DISK BLOCKS WRITTEN-------=         71   CRBUFOPT=MCYL
         DYNAMIC BUFFER RETRIEVALS-=          0    -    MAXIMUM BUFFERS USED------=        120
         MINIMUM BUFFERS USED------=          0    -    MAXIMUM BUFFERS AVAILABLE-=          0
         DATA SPACE USED(M BYTES)--=          1    -    DATA SPACE SIZE(M BYTES)--=        128

 IAM366  IAM COMMAND EXECUTION SUMMARY -
         GET RANDOM----------------=          0    -    PUT UPDATE----------------=          0
         GET SEQUENTIAL------------=          0    -    PUT ADD-------------------=      36000
         GET PREVIOUS--------------=          0    -    POINT (START BROWSE)------=          0
         GET KGE/GENERIC-----------=          0    -    POINT KGE/GENERIC---------=          0
         GET (SKIP SEQUENTIAL)-----=          0    -    ERASE---------------------=          0
         ENDREQ--------------------=          0    -    WRTBFR--------------------=          0
         IAM STATISTICS------------=          0    -    IAM FLUSH BUFFER----------=          0
         CLOSE---------------------=          1    -    OPEN----------------------=          1
         CLOSE TYPE=T--------------=          0    -    VERIFY--------------------=          0
         INVALID REQUESTS----------=          0    -    RECORD LENGTH CHANGES-----=          0
         SEQ CHAINED BLOCKS READ---=          0    -    SEQ CHAINED BLOCKS WRITTEN=       4122
```

Figure 56: Sample of an IAMINFO Report for a File Load

10.72 CONTINUED . . .

IAMINFO FOR FILE ACCESS A sample IAMINFO report for normal file access is shown below. A file access is indicated on the IAM361 message, where it will indicate either that the file is opened for INPUT or UPDATE processing. Other differences from the file load report include statistics for IAM's Dynamic Tabling feature, and statistics for the Index Space, which is a Data Space used to hold the index for the dataset.

```
IAM400 INNOVATION IAM INFORMATION PRINT ROUTINE--IAMNINFO VER 8.0/01P--INNOVATION DATA PROCESSING DATE-2002.169
 IAM360   STEP - VAR100F     DDNAME - VSAMCRT1    DATASET MONITORED - IAMV.VAR100.CLUSTER
 IAM361   INFO REQUESTED BY PROGRAM IAMTVSAM FOR UPDATE PROCESSING   OPENED-2002.169.13:50:43  CLOSED-2002.169.13:51:40

 IAM362   IAM DATA CHARACTERISTICS -                                                                  OVERRIDES IN EFFECT:
          IAM FILE FORMAT-----------=   ENHANCED    -    IAM FILE STATUS-----------=     LOADED
          LOGICAL RECORD LENGTH-----=       2040    -    CI SIZE-------------------=       2048
          KEY SIZE------------------=         58    -    KEY OFFSET----------------=          8
          TOTAL RECORDS-------------=      51000    -    TOTAL RECORDS DELETED-----=          0
          TOTAL RECORDS UPDATED-----=          0    -    TOTAL RECORDS INSERTED----=      15000

 IAM363   IAM FILE CHARACTERISTICS -
          BLOCKING FACTOR-----------=          4    -    BLOCK SIZE----------------=      13682
          TRACKS IN USE-------------=       1571    -    VARIABLE LENGTH OVERFLOW--=        YES
          DATASET TYPE--------------=       KSDS    -    SHARE OPTIONS-------------=          2
          NUMBER OF IAM DATA BLOCKS-=       4163    -    HIGH ALLOCATED RBN--------=      29999
          INTEGRATED OVERFLOW (CI%)-=         10    -    DASD RESERVE (CA%)--------=         10
          FILE DEFINED DATE---------=   2002.169    -   FILE DEFINED TIME---------=   13:47:30
          FILE LOADED DATE----------=   2002.169    -   FILE LOADED TIME----------=   13:47:53
          FILE UPDATE DATE----------=   2002.169    -   FILE UPDATE TIME----------=   13:50:42

 IAM372   IAM EXTENDED AREA CHARACTERISTICS -
          EXT. OVERFLOW RECORDS-----=      11831    -    EXT. OVERFLOW BLOCKS------=       1972
          EXTENDED BLOCKS ALLOCATED-=      25808    -    EXTENDED PE BLOCKS--------=          0
          EXTENDED BLOCKS USED -----=       2091    -    EXTENDED BLOCKS AVAILABLE-=      23717

 IAM365   IAM EXECUTION STATISTICS -
          TOTAL STORAGE REQUIRED----=     397312    -    PRIME INDEX(COMPRESSED)---=     138646
          STORAGE ABOVE THE LINE----=     389120    -    COMPRESSED DATA STRUCTURE-=        YES
          INDEX SPACE USED (K)------=       1152    -    TOTAL JOB INDEX SPACE USED=       3776   INDEXSPACE=YES
          REQUESTS PROCESSED--------=      51003    -    REQUESTS FAILED-----------=          1
          DISK BLOCKS READ----------=       2556    -    DISK BLOCKS WRITTEN-------=          0
          DYNAMIC BUFFER RETRIEVALS-=      26952    -    MAXIMUM BUFFERS USED------=         19
          MINIMUM BUFFERS USED------=          1    -    MAXIMUM BUFFERS AVAILABLE-=         19
          DYNAMIC TABLE RETRIEVALS--=          0    -    DYNAMIC TABLE RECORDS-----=          0

 IAM368   SPECIFYING A BUFNO VALUE GREATER THAN  19 MAY IMPROVE PERFORMANCE

 IAM366   IAM COMMAND EXECUTION SUMMARY -
          GET RANDOM----------------=          0    -    PUT UPDATE----------------=          0
          GET SEQUENTIAL------------=      51001    -    PUT ADD-------------------=          0
          GET PREVIOUS--------------=          0    -    POINT (START BROWSE)------=          0
          GET KGE/GENERIC-----------=          0    -    POINT KGE/GENERIC---------=          0
          GET (SKIP SEQUENTIAL)-----=          0    -    ERASE---------------------=          0
          ENDREQ--------------------=          0    -    WRTBFR--------------------=          0
          IAM STATISTICS------------=          0    -    IAM FLUSH BUFFER----------=          0
          CLOSE---------------------=          1    -    OPEN----------------------=          1
          CLOSE TYPE=T--------------=          0    -    VERIFY--------------------=          0
          INVALID REQUESTS----------=          0    -    RECORD LENGTH CHANGES-----=          0
          SEQ CHAINED BLOCKS READ---=       6163    -    SEQ CHAINED BLOCKS WRITTEN=          0
```

Figure 57: Sample IAMINFO Report for File Access

10.72 CONTINUED . . .

IAMINFO REPORT DESCRIPTION

The following tables will describe the various fields that appear on the IAMINFO Report. The first header line, IAM400 indicates the version and level of IAM that processed the dataset, and the date of the report. The second header line, IAM360, provides the Step Name, the DD Name of the file being processed, and the Dataset Name. The third header line, IAM361, indicates the name of the program that opened the IAM dataset, file processing mode (i.e. Creation, Input, or Update), and the open and close time stamps.

IAM362 DATA CHARACTERISTICS

IAM FILE FORMAT	Indicates if the file is an **ENHANCED** format IAM dataset, an SMS-EXT for DFSMD extended format, or a **COMPATIBLE** format IAM dataset.
IAM FILE STATUS	Indicates whether the file is in a **LOADED** state, or an **UNLOADED** state.
LOGICAL RECORD LENGTH	The maximum record length, as specified when the dataset was defined..
CI SIZE	The control interval (CI) size specified when file was defined. If no explicit CI size was provided, then IAM calculates a value based on the maximum record size.
KEY SIZE	The length of the key, specified when file was defined. The maximum key length for an IAM dataset is 249 bytes.
KEY OFFSET	The relative position of the key within the user data record. (RKP) The key must be positioned within the first 4K of the data record.
AVERAGE RECORD LENGTH	During a load operation IAM calculates an average length for the records that were loaded into the file. For data compressed files, the length is after compression. **NOTE:** This field only appears for a file CREATION report.
LARGEST RECORD PROCESSED	During a load operation IAM reports the length of the largest record that was written to the file. **NOTE:** This field only appears for a file CREATION report.
TOTAL RECORDS	The total number of records in the IAM file. **NOTE:** This field does not appear on a file CREATION report.
TOTAL RECORDS DELETED	The total number of records deleted from the IAM file since creation. **NOTE:** This field does notappear on a file CREATION report.
TOTAL RECORDS UPDATED	The total number of records updated in the IAM file since creation. **NOTE:** This field does not appear on a file CREATION report.
TOTAL RECORDS INSERTED	The total number of records inserted into the IAM file since creation. **NOTE:** This field does not appear on a file CREATION report.

10.72 CONTINUED . . .

IAM363 FILE CHARACTERISTICS

BLOCKING FACTOR A value of 1 to 15 is the number of blocks per track (as blocking factor). A larger value indicates the requested block size from the IAM CREATE B= override.

BLOCKSIZE Physical blocksize of the IAM file. This is the actual blocksize developed and used by IAM.

NOTE: IAM will determine the optimal blocksize for a file based upon the file's record length, the blocking factor requested and the track capacity of the device. The IAM blocksize is transparent to application programs even when the blocksize is altered or the file is relocated to a different device type.

TRACKS IN USE Number of tracks currently being used by IAM for the file. Does not include unused (available) Extended blocks.

VARIABLE LENGTH OVERFLOW A **YES** value indicates that IAM will fit as many records as possible into an extended overflow block. A **NO** value indicates that IAM will only put in the number of maximum size records that will fit within an overflow block.

DATASET TYPE Describes the type of dataset IAM is simulating. Possible values include: **KSDS**, for key sequenced dataset, **ESDS** for entry sequence, (i.e. sequential), **ESDS/P** for ESDS files defined with PSEUDORBA specified, or **AIX** for an alternate index.

SHARE OPTIONS Indicates the defined cross region share option for this file. Possible values are: **1, 2, 3,** or **4**. The use of IAM files with Share Option 3 or 4 is not recommended, as possible data loss can occur.

NUMBER OF IAM DATA BLOCKS The number of blocks in the file up to where the prime index begins. This field may be needed to run IAMRECVR for recovery if the first block in the file has been damaged.

HIGH ALLOCATED RBN This the highest allocated block number currently in the IAM file. This field may be needed to run IAMRECVR for recovery if the control information about the file has been damaged.

INTEGRATED OVERFLOW (CI%) The amount of space, as a percentage, left free in each prime data as the file is being loaded. The space can subsequently be used for file expansion. This value is specified and is similar in concept to CI Freespace.

DASD RESERVE (CA%) Indicates the CA Freespace value that was specified when the file was defined. This is used to by IAM to reserve some space for future expansion during automatic space release, which occurs during the first file load.

FILE DEFINED DATE / TIME The date and time that the file was defined.

FILE LOADED DATE / TIME The date and time of the completion of the last file load or reorganization for this file.

FILE UPDATE DATE / TIME The date and time the file was last closed by a program that updated the file.

10.72 CONTINUED . . .

IAM372 EXTENDED AREA CHARACTERISTICS

EXT. OVERFLOW RECORDS	Number of records currently contained in extended overflow blocks.
EXT. OVERFLOW BLOCKS	Number of extended overflow blocks.
EXTENDED BLOCKS ALLOCATED	Number of extended overflow blocks that will fit within the current amount of DASD space allocated to the file.
EXTENDED PE BLOCKS	Number of extended blocks assigned as Prime Extension (PE) blocks.
EXTENDED BLOCKS USED	Total number of extended overflow blocks in use. This includes Overflow, PE, and Extended Index blocks.
EXTENDED BLOCKS AVAILABLE	The number of extended blocks that are available for use within the currently allocated DASD space.

10.72 CONTINUED . . .

IAM365 EXECUTION STATISTICS

TOTAL STORAGE REQUIRED Total virtual storage from the job's address space acquired by IAM during this execution for this dataset.

PRIME INDEX [COMPRESSED] Indicates the amount of virtual storage required for the index to the prime data area of this file. Also, if the index is in a compressed format, the word COMPRESSED appears.

STORAGE ABOVE THE LINE Total amount of virtual storage acquired above the 16MB line.

COMPRESSED DATA STRUCTURE Indicates whether data compression was used for this file.

INDEX SPACE USED (K) The amount of virtual storage, in KB, used by this dataset in the Data Space that was acquired for IAM index structures.

TOTAL JOB INDEX SPACE USED The total amount of virtual storage, in KB, used so far by IAM in this job step in the Data Space that was acquired for the IAM Index Structure.

REQUESTS PROCESSED Number of requests made against the file, since last OPEN, by the application (a breakdown by command type follows in the command execution summary).

REQUESTS FAILED Number of requests made to IAM which did not complete normally (i.e. EOF on sequential read, no record found on random read)

DISK BLOCKS READ Number of physical I/O's used to read blocks from the file.

DISK BLOCKS WRITTEN Number of physical I/O's used to write blocks to the file, except during a load of a DFSMS extended format dataset, in which case this is will be the number of blocks written to disk.

DYNAMIC BUFFER RETRIEVALS Number of times that IAM was able to retrieve a block from the buffer pool without the need for a physical I/O.

MAXIMUM BUFFERS USED The maximum number of data buffers acquired by IAM's Real Time Tuning during this execution. This may be accompanied by an IAM367 or IAM368 informational message indicating that if IAM was allowed to acquire more buffers the number of I/O's required to service the requests against the file could have been reduced. All buffers are acquired above the 16MB line.

DYNAMIC TABLE RETRIEVALS Number of random read requests satisfied by IAM's Dynamic Tabling of data records in virtual. IAM's Dynamic Table (DYNCORE) is maintained in virtual storage above the 16MB line.

DYNAMIC TABLE RECORDS The total number of data records IAM placed into the Dynamic Table.

DATA SPACE USED (M BYTES) The amount of area actually used in the data space during file load.

DATA SPACE SIZE(M BYTES) The size of the data space created during file load.

10.72 CONTINUED . . .

IAM366 COMMAND EXECUTION SUMMARY

GET RANDOM	Number of random READ requests with an exact key specified.
PUT UPDATE	Number of UPDATE requests.
GET SEQUENTIAL	Number of sequential GET requests.
PUT ADD	Number of INSERT requests.
GET PREVIOUS	Number of GET PREVIOUS requests.
POINT (START BROWSE)	Number of POINT requests.
GET CI (SEQUENTIAL)	Number of sequential control interval GETS. (ESDS only)
PUT CI (UPDATE)	Number of CONTROL INTERVAL UPDATES (ESDS only)
GET CI (RANDOM)	Number of CONTROL INTERVAL GETS (ESDS only)
PUT CI (ADD)	Number of CONTROL INTERVAL ADDS (ESDS only)
GET KGE/GENERIC	Number of random or skip sequential reads, indicating search for key greater or equal, and /or partial key search. (KSDS only)
POINT KGE/ GENERIC	Number of POINT, or START BROWSE, requests indicating a search for key greater or equal, or partial key search. (KSDS only)
GET (SKIP SEQUENTIAL)	Number of GET's issued in Skip Sequential mode, with an exact key specified. (KSDS only)
ERASE	Number of requests to delete records. (KSDS only)
ENDREQ	Number of LSR ENDREQ requests
WRTBFR	Number of LSR WRTBFR requests
IAM STATISTICS	Number of IAM statistic requests
IAM FLUSH BUFFER	Number of IAM flush buffer requests
CLOSE	Number of CLOSE requests
OPEN	Number of OPEN requests
CLOSE TYPE=T	Number of TEMPORARY CLOSE requests
VERIFY	Number of VERIFYs
INVALID REQUESTS	Number of requests with RPL error's
RECORD LENGTH CHANGES	Number of records with length changes
SEQ CHAINED BLOCKS READ	Number of chained buffers during READ I/O.
SEQ CHAIN WRITTEN	Number of chained buffers during WRITE I/O.

10.72 CONTINUED . . .

IAMINFO REPORTS FROM IAMSMF

As indicated previously, if the IAM Global Options have been set to enable the recording of IAM SMF records, then the IAMSMF program can be used to print IAMINFO reports from the SMF data. This is a handy way to make sure that the information will be available, without having to put IAMINFO DD cards in all of the jobs using IAM datasets. IAMSMF can be run either against the live SMF dataset, or SMF history datasets. To obtain the reports, use the IAMINFO command of IAMSMF.

The IAMINFO command offers a variety of selection criteria, so that you can limit the number of reports produced, if desired. Amongst the criteria that can be specified are job names, dataset names, by dates, or by various dataset attributes or activity levels. Full details on the various options and keywords are available in the System Analysis Utilities section of the manual.

EXAMPLE OF IAMSMF JCL FOR IAMINFO REPORTS

An example of running the IAMSMF utility to print IAMINFO reports is shown below. The DD cards required are SYSPRINT for the printed report(s), SYSIN for the control card input, and SYSMF which is the dataset containing the SMF data. In the example, IAMSMF will print out IAMINFO reports only for the specified job name, plus MERGE=NO means that a report is produced for each IAM dataset close.

```
//IAMINFO   EXEC PGM=IAMSMF,REGION=0M
//SYSPRINT  DD   SYSOUT=*
//SYSMF     DD   DISP=OLD,DSN=my.smf.data
//SYSIN     DD   *
   IAMINFO   JOBNAME=myjob,MERGE=NO
/*
```

Figure 58: Example JCL for IAMIFNO Reports from IAMSMF (EX1072B)

The IAMINFO reports that are produced by IAMSMF have the IAM360 and IAM361 messages replaced by an IAM370 Job Characteristics Section. The fields in that section are described below.

IAM370 JOB CHARACTERISTICS

JOB NAME Indicates the name of the job that processed the IAM dataset.

STEP NAME Indicates the job step name that processed the IAM dataset.

PROGRAM NAME The name of the program that opened the IAM dataset.

FUNCTION Indicates the function, which can be File Creation, Input Processing, or Update Processing.

DDNAME The DD name that was processed.

DSNAME The name of the IAM dataset processed.

DATE OPENED The date that the dataset was opened.

TIME OPENED The time that the dataset was opened.

DATE CLOSED The date that the dataset was closed.

TIME CLOSED The time that the dataset was closed.

10.73 IAMSMFVS REPORTS

IAMSMFVS Another reporting alternative, if the IAM SMF records are being recorded, is to use the IAMSMFVS program. IAMSMFVS provides a handy way to monitor and track the use of IAM within an installation, or by particular applications. This program produces three reports, an EXCP report, a Dataset Report, and a Size Report. Each report summarizes activity with one line per dataset and reports on different statistics and attributes. The EXCP report and the Size report by default report on the top 100 datasets, while the Dataset Report contains all of the datasets that were processed. Some customers have used the IAMSMFVS report to watch and determine when various IAM datasets may need to be reorganized based on percentage of the overflow area in use.

EXAMPLE IAMSMFVS JCL An example of the JCL and control card to run IAMSMFVS are shown below. Full details on running IAMSMFVS and on the control card input are provided in the System Analysis Utilities section of the manual. Please note that IAMSMFVS requires SMF type 30, subtype 4 records or SMF type 4 records to produce reports. There is one keyword specified on the example REPORT control card, which is DSORG=IAM. By default, IAMSMFVS will produce reports on IAM and VSAM. (Note: For VSAM only, specify DSORG=AM).

```
//IAMSMFVS   EXEC  PGM=IAMSMFVS,REGION=0M
//SYSMF       DD   DISP=SHR,DSN=my.smf.data
//SYSPRINT    DD   SYSOUT=*
//SORTIN      DD   UNIT=SYSDA,SPACE=(CYL,(15,5))
//SORTOUT     DD   UNIT=SYSDA,SPACE=(CYL,(15,5))
//SORTWK01    DD   UNIT=SYSDA,SPACE=(CYL,(15,10))
//SORTWK02    DD   UNIT=SYSDA,SPACE=(CYL,(15,10))
//SORTWK03    DD   UNIT=SYSDA,SPACE=(CYL,(15,10))
//SYSOUT      DD   SYSOUT=*
//SYSIN DD *
   REPORT  DSORG=IAM
/*
```

Figure 59: Example of JCL to run IAMSMFVS (EX1073A)

A few additional keywords for the REPORT control card that you might find useful for IAM reporting include:

- **CURRENT –** Specifies that the current overflow usage values be reported on, rather than the maximum amount encountered in the data.
- **DETAIL –** The EXCP report will contain a breakdown for each use of the file, including job, step, and program name information.
- **DSG –** Specifies only datasets that begin with the value(s) specified will be included in the reports. For example, DSG=PROD, or DSG=(PROD1,PROD2.AP,PROD3.AR)
- **MAXDSNS –** Specifies the maximum number of datasets that IAMSMFVS will accumulate statistics for. The default value is 1500. Note that for VSAM, each component counts as a dataset, plus the cluster name. So, a KSDS will require 3 entries, and an ESDS will require 2 entries. IAM datasets only require 1 entry each.
- **MAXREPORTS –** Specifies the number of datasets that will be included in the EXCP and the SIZE reports. Default is 100.

10.73 CONTINUED . . .

SAMPLE IAMSMFVS SUMMARY REPORT

The first page of the IAMSMFVS output consists of a summarization of all of the data that was selected for reporting, including totals for dataset names, job names, total EXCP counts by access method, and DASD space used by access method (i.e., IAM and VSAM). Below is sample output of the first page.

```
IAM400 SMF REPORT/DATA EXTRACT PROGRAM-IAMSMFVS VER 8.0/01P-INNOVATION DATA PROCESSING DATE-2002.169
IAM303   CARD IMAGE - *   REPORT  DSORG=IAM        00120035*
IAM491   SMF REPORT         FUNCTION STARTED - 08.54.31

IAM601 SMF RECORDS -- READ......4763  USED.......392  DROPPED.........0
        22 DATASETS TABLED REPRESENTING     2 JOBS --  DATED  2002.169 07:38  THRU  2002.169 08:31
          SPACE UTILIZATION SUMMARY -

                DEVICE TYPE.....3380    VSAM CYL/TRK........0/00  IAM CYL/TRK......139/02
                DEVICE TYPE.....3390    VSAM CYL/TRK........0/00  IAM CYL/TRK......274/08
          TOTAL DISK EXCPS.......36336  VSAM EXCPS.............0  IAM EXCPS.........30112
```

Figure 60: Sample IAMSMFVS Summary Report

SAMPLE IAMSMFVS IAM EXCP REPORT

After the Summary Report, IAMSMFVS produces the EXCP report. There is a separate EXCP report for IAM datasets and VSAM datasets, if both types of datasets are being reported on. In our sample case, only IAM datasets are being reported on, so there is only an IAM EXCP report. Below is a sample of an IAM EXCP report, as produced by IAMSMFVS. Entries are sorted by largest to smallest EXCP count. As stated previously, only the top 100 datasets are included, unless MAXREPORTS has been specified with a different value.

```
IAM400 SMF REPORT/DATA EXTRACT PROGRAM-IAMSMFVS VER 8.0/01P-INNOVATION DATA PROCESSING DATE-
2002.169
                                    IAM EXCP REPORT
                           USE     TOTAL                                                          OVRFLW   USED
 DATASET NAME             COUNT     EXCPS   RECORDS     READS INSERTS  UPDATES DELETES   RECS   %   TRKS
 ---- --- ----             -----     -----   -------     ----- -------  ------- -------   ---- ---   ----
 IAMV.DCV209.CLUSTER          2      9923     40960    164867       0     1024       0    140  10    704
 IAMV.KSD985.CLUSTER          7      7031    167213    160022    3630        0       0   3606  11   4118
 IAMV.DCV207.CLUSTER          2      3490      8192     34819       0     2048       0    252  20     80
 IAMV.DCV205.CLUSTER          2      2628      5824     19139     864     1024       0    449  33    100
 IAMV.DCV211.CLUSTER          2      2344     20000     28193       0        0       0      0 100    346
 IAMV.DCV202.CLUSTER          2      1674      5120     22083     512       64       0    256  11     47
 IAMV.DCV204.CLUSTER          2      1309      5824     19139     864     1024       0    462  17     51
 IAMV.DCV203.CLUSTER          3       752      6172     11794    1038     1524       0    655  35     24
 IAMV.DCV210.CLUSTER          2       381    100000    108193       0        0       0      0 100    228
 IAMV.DCV208B.CLUSTER         3       185     10800     21601       0        0       0      0 100     41
 IAMV.DCV201.CLUSTER          2       126      4096     17411       0     1024       0     79   1      7
 IAMV.DCV213.CLUSTER          2       116     10000       192       0       60       4      0 100    178
 IAMV.DCV208A.CLUSTER         3        46     10800     21602       0        0       0      0 100     49
 IAMV.DCV212.CLUSTER          1        25     10000         0       0        0       0      0 100    178
 IAMV.DCV402.CLUSTER          3        18        42         9       4        0       0      0   0      4
 IAMV.DCV206.CLUSTER          2         8      4096      4097       0        0       0      0 100      6
         END OF IAM EXCP REPORT
```

Figure 61: Sample IAMSMFVS IAM EXCP Report

The Use Count is the number of SMF records processed for the dataset. The following fields represent the accumulation of data from all of the records: EXCPS, READS, INSERTS, UPDATES, and DELETES. If the keyword CURRENT is included on the REPORT request, then the Overflow statistics are from the most recent record, otherwise they are the maximum encountered. The RECORDS and TRACKS USED are the maximum encountered.

10.73 CONTINUED . . .

SAMPLE IAMSMFVS DATASET REPORT

The next report is the Dataset Summary Report. This report is sorted by dataset name, and includes all of the datasets selected for this execution of IAMSMFVS. The information presented includes basic dataset attribute information, such as record length, key length, relative key position, and the block size or CI size.

```
IAM400 SMF REPORT/DATA EXTRACT PROGRAM-IAMSMFVS VER 8.0/01P-INNOVATION DATA PROCESSING DATE-2002.169
                                DATASET SUMMARY REPORT
                                USE      TOTAL                          AVG     MAX    KEY          BLK OR    FRSPC
 DATASET NAME                  COUNT     EXCPS   DSORG   RECFM   LRECL   LRECL   LEN    RKP   CISIZE   CI% CA%
 ---- --- ----                  -----     -----   -----   -----   -----   -----   ---    ---   ------   --- ---
 IAMV.DCV201.CLUSTER              2        126    IAM     VB-DC      46      64    4     12    11476    10  10
 IAMV.DCV202.CLUSTER              2       1674    IAM     VE-DC     429     670    4      8    11476     6  10
 IAMV.DCV203.CLUSTER              3        752    IAM     VO-DC     166     256    4      8    11476    10  20
 IAMV.DCV204.CLUSTER              2       1309    IAM     VO-DC     418     512    4    252    11476    10  10
 IAMV.DCV205.CLUSTER              2       2628    IAM     VE-DC     838    1024    4    516    11476    10  10
 IAMV.DCV206.CLUSTER              2          8    IAM     VE-DC      46      64    4     12    11476    10  10
 IAMV.DCV207.CLUSTER              2       3490    IAM     VB-DC     350    1040   16      8    11476    10  10
 IAMV.DCV208A.CLUSTER             3         46    IAM     VE        175     175   75      8    11476    10  10
 IAMV.DCV208B.CLUSTER             3        185    IAM     VB-DC     147     175   75      8    11476    10  10
 IAMV.DCV209.CLUSTER              2       9923    IAM     VB-DC     669    1040   24      8    11476    10  10
 IAMV.DCV210.CLUSTER              2        381    IAM     VB-DC      89     128   12     12    11476    10  10
 IAMV.DCV211.CLUSTER              2       2344    IAM     VB-DC     677    1039   12     12    11476    10  10
 IAMV.DCV212.CLUSTER              1         25    IAM     VB-DC     704    1024   12     12    11476    10  10
 IAMV.DCV213.CLUSTER              2        116    IAM     VE-DC     704    1024   12     12    11476    10  10
 IAMV.DCV402.CLUSTER              3         18    IAM     VE-DC    4680    2340    4      8    23476    10  10
 IAMV.KSD985.CLUSTER              7       7031    IAM     VB       1300    1300   60      8    13682     0
```

Figure 62: Sample IAMSMFVS Dataset Report

The RECFM values for IAM datasets have the following meanings:

FB: Compatible format, fixed length records

VB: Compatible format, variable length records

VE: Enhanced format, variable length records

VO: Enhanced format, variable overflow

DC: Data Compressed

Note that for IAM datasets, the average record length is based only on records initially loaded into the dataset, and if data compression was being used, will be based on the record lengths after compression.

10.73 CONTINUED . . .

SAMPLE IAMSMFVS IAM SIZE REPORT

The Size report is broken out in a manner similar to the EXCP report. There are separate reports for IAM and VSAM datasets, and the reports consist of the largest 100 datasets, or whatever value was specified on the MAXREPORTS operand. This report contains the most information about the overflow area use, so it would be the one to use for a determination as to whether reorganization is needed.

IAM400 SMF REPORT/DATA EXTRACT PROGRAM-IAMSMFVS VER 8.0/01P-INNOVATION DATA PROCESSING DATE-2002.169
IAM SIZE REPORT

DATASET NAME	TRACKS USED	TOTAL EXCPS	USE COUNT	TOTAL RECORDS	INDEPENDENT OVERFLOW MAX REC	INDEPENDENT OVERFLOW USE REC	INDEPENDENT OVERFLOW % USE	PRIME EXT	CI%
IAMV.KSD985.CLUSTER	4118	7031	7	167213	30290	3606	11	0	0
IAMV.DCV209.CLUSTER	704	9923	2	40960	1400	140	10	0	10
IAMV.DCV211.CLUSTER	346	2344	2	20000	0	0	0	0	10
IAMV.DCV210.CLUSTER	228	381	2	100000	0	0	0	0	10
IAMV.DCV213.CLUSTER	178	116	2	10000	0	0	0	0	10
IAMV.DCV212.CLUSTER	178	25	1	10000	0	0	0	0	10
IAMV.DCV205.CLUSTER	100	2628	2	5824	1342	449	33	7	10
IAMV.DCV207.CLUSTER	80	3490	2	8192	1260	252	20	0	10
IAMV.DCV204.CLUSTER	51	1309	2	5824	2640	462	17	4	10
IAMV.DCV208A.CLUSTER	49	46	3	10800	0	0	0	0	10
IAMV.DCV202.CLUSTER	47	1674	2	5120	2176	256	11	0	6
IAMV.DCV208B.CLUSTER	41	185	3	10800	0	0	0	0	10
IAMV.DCV203.CLUSTER	24	752	3	6172	1848	655	35	4	10
IAMV.DCV201.CLUSTER	7	126	2	4096	5712	79	1	0	10
IAMV.DCV206.CLUSTER	6	8	2	4096	0	0	0	0	10
IAMV.DCV402.CLUSTER	4	18	3	42	227	0	0	4	10

Figure 63: Sample IAMSMFVS IAM Size Report

For Enhanced Format files, the Maximum Overflow Records are based on either the user specified number of overflow records, from the CREATE O= override. The O= override value is retained for informational purposes to aid in making decisions about when the dataset should be reorganized. IAM makes no guarantee that there is sufficient DASD space for IAM to actually keep that number of records in the overflow area. Also, the overflow may be able to hold more records than the number specified, so the overflow used percentage can exceed 100.

If no override had been provided, then maximum overflow records is calculated based on the number of extended blocks currently allocated to the dataset.

10.74 REPORTING ON IAM DATASETS WITH FDREPORT

FDREPORT Customers who have the ABR product from Innovation Data Processing can use FDREPORT to find and provide information on IAM datasets. FDREPORT is a very powerful and flexible reporting tool, that enables the user to customize reports with information about datasets residing on their DASD volumes. FDREPORT can pull information from the VTOC, the catalog, the VVDS, and for IAM datasets, statistical and attribute information from the dataset itself, and merge that information into a single report. Using FDREPORT, you can find out where all of your IAM datasets are, which datasets are IAM datasets, how much space is being used by IAM, does a particular volume or volume group have any IAM datasets, plus many dataset management other questions.

Full information on using FDREPORT is in Section 54 of the FDR manual. It is quite easy to use. First, you specify what datasets you want to report on using the SELECT or XSELECT cards. You can also exclude various datasets with the EXCLUDE or XEXCLUDE control cards. Then, using the REPORT card, you specify what fields you want included in your report. Next, you can provide your own report title using the TITLE card. Then, with the PRINT card, FDREPORT goes to work for you, and very quickly you will have your own customized report.

EXAMPLE 1: FDREPORT JCL AND CONTROL CARD Shown below is a sample of the JCL and control cards that can be used to find all of the IAM datasets in your installation. The information being requested includes:

1. **SPLDSN** - The dataset name, split across two lines if it is very long.
2. **VOL** - The volume containing the dataset.
3. **SIZE** - The space allocated to the dataset, in tracks.
4. **%FREE** - The percentage of unused space.
5. **NOEXTENT** - The number of extents.
6. **BLKSIZE** - The block size of the dataset.
7. **MAXLRECL** - The maximum record length.
8. **KEYLEN** - The key length of the dataset.
9. **RKP** - The relative key position.
10. **RECORDS** - The number of records in the dataset.
11. **OVERUSED** - The number of records in the IAM Overflow Area

```
//KSD972C  EXEC PGM=FDREPORT,REGION=0M
//SYSPRINT  DD  SYSOUT=*
//SYSIN     DD  *
 XSELECT DSORG=IAM
 REPORT FIELDS=(SPLDSN,VOL,SIZE,%FREE,NOEXTENT,
               BLKSIZE,MAXLRECL,KEYLEN,RKP,RECORDS,OVERUSED)
 TITLE LINE='IAM DATASET REPORT'
 PRINT ENABLE=IAM,SORT=COMBINE,SORTALLOC=(SORTWORK,CYL),ONLINE
/*
```

Figure 64: Sample JCL to run FDREPORT (EX1074A)

10.74 CONTINUED . . .

SAMPLE FDREPORT OUTPUT

The report produced by the JCL and control cards above looks like this:

```
FDR400 FDRABR CUSTOM REPORTS - FDREPORT VER 5.2/65P INNOVATION DATA PROCESSING DATE-97.269 PAGE-0001
                                       IAM DATASET REPORT

                                                                  KEY
 DATASET NAME                VOLSER  ALLOC %FR EXT BKSIZ  MAXLR LEN  RKP    RECORDS   OVERUSED
 -------------               ------ ------ --- --- -----  ------ ---  ---- ---------- ----------
 GFM.P.EMBOX                 IDPLB5   7500   0   3 32760  32750  55    4     651727       5178
 IAMV.$IAM.GAMA.KSDS2        IDPLB2     15   0   1 13682    500   8    4        500        100
 IAMV.$IAM.GAMA0008.CLUSTER  IDPLB2     15  86   1 13682     80   8    4        185          0
 IAMV.$IAM.GAMA8.CLUSTER     IDPLB2     15  86   1 13682     80   8    4        185          0
 IAMV.KSD972A.CLUSTER        SCR092    750  79   1 13682   1300   4   12       6000          0
 IAMV.KSD972B.CLUSTER        SCR092    750  86   1 13682   1300   4   12       4000          0
 IAMV.KSD972C.CLUSTER        SCR092    750  69   1 13682   1300   4   12       9000          0
 IAMV.VIT212A.CLUSTER        SCR083     30  46   1 23476   2340   4   12        256          0
 MET1.MAILFILE               SCR083    960   0  12 11476   2260   8    4      42587          0
 SYSP.USTPROD.$USTCAT        IDPLB4    150  76   1 11476    504  18    4        686          0
 SYSP.USTPROD.$USTINFO       IDPLB4   6900   4   5 15476    354  50    4    1557834          0
```

Figure 65: Sample Report from FDREPORT

EXAMPLE 2: FDREPORT SELECTIVE CRITERIA

A more selective example is shown below. For this report request, selection criteria has been added to look only for datasets that begin with the specified character string, as identified by the DSG= parameter, that are on volumes that begin with the character string identified by VOLG=.

```
//KSD972C  EXEC PGM=FDREPORT,REGION=0M
//SYSPRINT  DD  SYSOUT=*
//SYSIN     DD  *
 XSELECT DSORG=IAM,DSG=prod.ordent,VOLG=PROD
 REPORT FIELDS=(DSN,VOL,SIZE,NOEXTENT,IAMINFO,KEYLEN)
TITLE LINE='IAM ORDER ENTRY DATASET REPORT'
 PRINT ENABLE=IAM,SORT=COMBINE,SORTALLOC=(SORTWORK,CYL),ONLINE
/*
```

Figure 66: Example FDREPORT for IAM Datasets (EX1074B)

An example of the output is shown below.

```
FDR400 FDRABR CUSTOM REPORTS-FDREPORT VER 5.2/65P INNOVATION DATA PROCESSING DATE-97.272 PAGE - 0001
                                  IAM ORDER ENTRY DATASET REPORT

                                                 D/S                                      KEY
 DATASET NAME                  VOLSER  ALLOC EXT ORG RECFM  MAXLR LRECL BKSIZ CISIZ LEN
 -------------                 ------ ------ --- --- -----  ------ ----- ----- ----- ---
 PROD.ORDENT.MEPFILE           IDPLB5     15   1 IAM FB        16    16 11472     0   6
 PROD.ORDENT.NAMEFILE          IDPLB2     60   1 IAM FB        99    99 13662     0  69
 PROD.ORDENT.PRICE.FILE        IDPLB2      5   1 IAM FB        30    30 13680 11460  14
 PROD.ORDENT.PROD.LEVEL        IDPLB2     77   2 IAM FB        17    17 13668 11475  13
 PROD.ORDENT.FILE              IDPLB2   1026   3 IAM VB      2260   810 13682 11476   8
```

Figure 67: Example 2 of FDREPORT Output

10.80 IAM DATASET MAINTENANCE

OVERVIEW As with any indexed dataset, IAM datasets require periodic maintenance. This includes such chores as periodic reorganizations, renaming, recataloging, backup and restore, moving, and on rare occasion a recovery may be necessary. This section will explain why and when such chores are necessary, and provide plenty of examples on how to accomplish each chore to meet your processing requirements.

10.81 REORGANIZING IAM DATASETS

WHY REORGANIZE

With the enhancements made to IAM in prior versions, datasets may now be able to exist for quite a long length of time before reorganization is necessary due to increased DASD storage requirements for the Overflow area. However, as with any indexed dataset type, IAM datasets that are updated will have to be periodically reorganized for optimum performance and DASD space usage. As records are added or updated with record length increases, these records are placed into the Extended areas (Extended Overflow / Extended PE) of the IAM dataset. This causes the amount of virtual storage required to index the file to increase. As the overflow index grows, there will also be additional CPU time required to open the dataset, and to search and maintain position within the index. Additionally, for sequential processing, keys in ascending key sequence can end up scattered throughout the extended overflow area causing excessive I/O when reading or updating the dataset. Datasets are still subject to running out of extents or DASD space. Deleted records may leave significant portions of the prime area empty and not reusable because the original prime index is no longer reflective of the data in the file. For these reasons, updated files are going to need periodic reorganization.

WHEN TO REORGANIZE

Many applications that have been using VSAM or IAM datasets already have regularly scheduled dataset reorganizations. Generally, there should be no need to change the schedule. For datasets that are only reorganized when needed, the difficulty is in determining when an IAM dataset should be reorganized. With Compatible format IAM files, it was relatively easy because of the fixed size of the overflow area. With Enhanced format files the ability to acquire extents and now with variable overflow, it is not quite as clear cut. Amongst the criteria to consider are:

1. Amount of virtual storage required for the Extended Overflow Index. This can be estimated by multiplying the number of records in Overflow by the key length plus 4.
2. Percentage of records that are in Extended Overflow. For example, if more than 10% of the records in the dataset are in Extended Overflow, then the dataset should be reorganized.
3. Number of extents being used for single volume dataset.
4. The Extended area of the file exceeding a specified quantity of DASD space.
5. Exceeding specified quantity of records in Extended Overflow.

To aid in determining when a dataset should be reorganized, IAM will produce a warning message, IAMW22, during open or close if certain conditions are detected for which a reorganization might be warranted. These messages are meant to bring to attention that a reorganization should be considered for the indicated datasets. Certainly some datasets may continue processing for several days or weeks without a reorganization after hitting a threshold condition, but only experience will be able to indicate if that is true. Other datasets, such as small datasets with very high activity, may need to be reorganized more frequently. The IAM thresholds include the Extended Overflow index exceeding sixteen megabytes of storage, using thirteen or more extents for a single volume dataset, or when the overflow area exceeds one thousand cylinders. The Overflow override can be used to monitor how much overflow space is being used, similar to what was done for Compatible format IAM files, except that the limit is no longer the absolute limit. By providing the override, IAM will monitor and inform you via an IAMW22 message that a reorganization is recommended because the Extended Overflow area exceeds the number of records specified.

Reorganization can be done automatically. Innovation Data Processing offers a software package, FDRREORG, which can automate file reorganizations for IAM, VSAM, and PDS types of datasets. Other alternatives used by some customers include using the IAM SMF reports or the IAM SMF records directly to trigger dataset reorganizations, and the IAMW22 message has also been used. Also the IAMPRINT report produced by a LISTCAT can trigger a dataset reorganization.

NOTE: FDRREORG must be at the 5.4/20 level or higher for IAM data sets that use hardware compression and/or are DFSMS extended format.

10.81 CONTINUED . . .

HOW TO REORGANIZE

IAM datasets can be easily reorganized with the FDRREORG product from Innovation Data Processing. FDRREORG can search out all or just the selected IAM datasets, and if they meet the specified criteria, will be automatically reorganized. FDRREORG maximizes buffering for fast reorganizations, can reorganize multiple datasets concurrently, and has a feature to DELETE and DEFINE the datasets that are being reorganized. FDRREORG provides the capability to read from each volume of a multivolume IAM dataset concurrently (the MAXPARALLELBACKUPS= keyword), which can result in faster reorganization times for the very large datasets.

The other common way to reorganize an IAM dataset is with IDCAMS REPRO. With appropriate buffering, IDCAMS REPRO can perform the reorganization quickly. The general technique is to REPRO the data from the IAM dataset into a sequential dataset, then optionally delete and redefine the IAM dataset, followed by an IDCAMS REPRO REUSE from the sequential dataset into the IAM dataset. An alternative with IDCAMS is to REPRO from one IAM dataset into another IAM dataset without going to a sequential dataset. This could save considerable time, providing that there is the DASD space available for both copies of the dataset, and that a sequential back up of the data is not required. Some installations have written programs to read the IAMSMFVS report for IAM datasets or to read the optional IAM SMF records directly, and then build and submit job(s) to reorganize the selected files. The IAMPRINT LISTCAT output has also been used for this purpose.

Some application software packages provide their own dataset reorganization utility. The one thing to watch out for with such utilities is to determine if they are doing a single record load followed by a mass insert. Such a process will result in the data records all being placed within the extended areas of the file, which may adversely impact the performance of the file in subsequent executions. If such is the case, then an additional reorganization by IDCAMS should be done following the application reorganization to ensure optimum performance.

ALTERNATE INDEX CONSIDERATIONS

IAM base cluster types of datasets with alternate indexes are marked as non-reusable by IAM, same as with VSAM. The base cluster cannot be reorganized without deleting and redefining the base, and all of the related components except by FDRREORG. This is to help insure that the alternate index files retain their integrity with the base cluster. The IAM alternate index datasets can be reorganized as needed, or they can just be rebuilt. FDRREORG V5.3/50 or higher is required to reorganize IAM Alternate Index datasets, or FDRREORG V5.4/20 if any of the IAM datasets are hardware compressed or in a DFSMS extended format.

BACKUP OF COMPRESSED DATA

An IAM feature that may help speed up the reorganization is to not decompress the data while it is being read, and to accept the IAM compressed data format on the reload. This feature will eliminate the CPU time spent decompressing and compressing the data, and eliminate I/O transfer time to the output device. If the output is a tape device that offers compression, there normally will not be any savings in tape usage because the data is already compressed. FDRREORG will automatically invoke this facility. For IDCAMS REPRO, the IAM ACCESS and CREATE Override of BACKUPCOMPRESSED must be specified on both the backup and the reload portion of the reorganization. Note that the sequential file record length must be at least eight bytes larger than the defined maximum record size for the dataset being copied, and should also have a record format of variable blocked (RECFM=VB). This facility is not recommended for use when the sequential backup is read by other application programs, as the data is not in a usable form. The compressed backup file can be converted to an uncompressed sequential file by the IAMRECVR DECOMPRESS command. For the adventuresome, IAM provides a callable service that will read and decompress a sequential file with compressed IAM data, and can also write out user data to a sequential file in IAM compressed format. Refer to the Reference Section of this manual for further information on using the callable service. Note that while SYNCSORT can be used to reorganize IAM files, it cannot use the BACKUPCOMPRESSED feature because of the mechanisms it uses to determine the file characteristics.

10.81 CONTINUED . . .

EXAMPLE A: MULTIPLE IAM DATASET REORG WITH FDRREORG

The first example of reorganizing IAM datasets is with FDRREORG. In this example, all cataloged IAM datasets that have a high level index of MYIAM will be considered for reorganization. The criteria being specified is that if any of the following are true, the dataset will be reorganized:

- The Overflow Index exceeds 4 megabytes.
- More than 10% of the records in the file are in the Extended Overflow area.
- There are more than 1,000,000 records in Overflow.

Defaults for the backup datasets and for the maximum tasks will be used.

```
//REORGIAM EXEC  PGM=FDRREORG,REGION=0M
//SYSPRINT  DD   SYSOUT=*
//REORGPRT  DD   SYSOUT=*
//REORGRPT  DD   SYSOUT=*
//IAMINFO   DD   SYSOUT=*
//SYSIN     DD   *
  REORG DSTYPE=IAM
  SELECT CATDSN=(MYIAM.**),IFANY,
      OVERFLOWINDEX>4194304,
      PCTTRECO>10,
      ORECS>1000000
/*
```

Figure 68: Reorganization of Multiple IAM Datasets Selected by FDRREORG (EX1081A)

EXAMPLE B: PARALLEL VOLUME I/O WITH FDRREORG

In this example, a single multi-volume IAM dataset is reorganized in parallel mode. This feature allows concurrent reading of the different volumes on which the IAM dataset resides, which can reduce reorganization time depending on the I/O configuration for both the source IAM dataset, and the backup datasets. Note that as required by FDRREORG, the BACKUPINDEX is specified with a ? embedded so that each volume backup has a unique name.

```
//REORGIAM EXEC  PGM=FDRREORG,REGION=0M
//SYSPRINT  DD   SYSOUT=*
//REORGPRT  DD   SYSOUT=*
//REORGRPT  DD   SYSOUT=*
//IAMINFO   DD   SYSOUT=*
//SYSIN     DD   *
  REORG MODE=P,MAXP=4,BACKUPI=++BACKUP?
  SELECT CATDSN=(MYIAM.DATASET)
/*
```

Figure 69: Reorganization with Multiple Volume Parallel Read (EX1081B)

10.81 CONTINUED . . .

EXAMPLE C: IDCAMS REORG WITH DELETE AND DEFINE

In this example, the IAM dataset is reorganized with an IDCAMS REPRO. IAM Overrides are used to maximize buffering, and to not decompress the data. The sequential dataset is being written to a 3390 type of DASD volume, where _ track blocking is being specified. Note that a permanent dataset name is being given for the sequential file, and that the dataset is being kept and cataloged, even if the job abends. It can be deleted once the reorganization has been verified as successful. Extreme caution must be used to prevent the loss of valuable data. Note the use of the IF MAXCC conditional operation under IDCAMS, which will stop the reorganization if any errors occur. This is done to provide maximum protection for the data being reorganized.

Warning: Do NOT use a temporary dataset for the sequential output file! Doing so may result in data loss should the file reload portion of the REPRO fail for any reason!

Note that the IAM dataset is referenced by the IDS and ODS parameters. This will cause IDCAMS to dynamically allocate the IAM dataset, which is being done just in case the define is changed at some time to place the IAM dataset on a different volume.

```
//REORG     EXEC  PGM=IDCAMS,REGION=0M
//SYSPRINT   DD    SYSOUT=*
//IAMINFO    DD    SYSOUT=*
//IAMPRINT   DD    SYSOUT=*
//BACKUP     DD    DSN=my.backup.iam.dataset,
//                 DISP=(NEW,CATLG,CATLG),
//                 UNIT=3390,VOL=SER=MY3390,
//                 SPACE=(CYL,(100,50),RLSE),
//                 DCB=(RECFM=VB,LRECL=108,BLKSIZE=27998,BUFNO=30)
//IAMOVRID   DD    *
   ACCESS    DD=&ALLDD,MINBUFNO=32,MAXBUFNO=64,BACKUPCOMPRESSED
   CREATE    DD=&ALLDD,CRBUFOPT=MCYL,BACKUPCOMPRESSED
/*
//SYSIN      DD    *
  LISTCAT  ENT(my.iam.dataset) ALL
  REPRO IDS(my.iam.dataset) OUTFILE(BACKUP)
  IF MAXCC NE 0 THEN CANCEL
  DELETE my.iam.dataset CLUSTER
  IF MAXCC NE 0 THEN CANCEL
  DEFINE CLUSTER                      -
     (NAME(my.iam.dataset)            -
            OWNER($IAM)               -
            VOL(myvol)                -
            CYL(100 50)               -
            RECSZ(64 100)             -
            KEYS(16 0)                -
            FREESPACE(5 10)           -
            SHAREOPTIONS(2 3)         -
            REUSE        )
  IF MAXCC NE 0 THEN CANCEL
  REPRO  INFILE(BACKUP) ODS(my.iam.dataset)
  LISTCAT ENT(my.iam.dataset) ALL
/*
```

Figure 70: Example of using IDCAMS to reorganize an IAM dataset.

10.81 CONTINUED . . .

EXAMPLE 4: IDCAMS REORG WITH REUSE

In this next IDCAMS example, the IAM dataset is reorganized without doing a DELETE and DEFINE. Because of that, the IAM dataset can be specified in JCL without any concerns. Note in this example that the keyword REUSE has to be specified when reloading the dataset. If this is not done, then IDCAMS does an update processing instead of load processing, and the dataset will not be reorganized.

```
//REORG     EXEC  PGM=IDCAMS,REGION=0M
//SYSPRINT   DD    SYSOUT=*
//IAMINFO    DD    SYSOUT=*
//IAMPRINT   DD    SYSOUT=*
//MYIAMDS    DD    DSN=my.iam.dataset,DISP=OLD
//BACKUP     DD    DSN=my.backup.iam.dataset,
//                 DISP=(NEW,CATLG,CATLG),
//                 UNIT=3390,VOL=SER=MY3390,
//                 SPACE=(CYL,(100,50),RLSE),
//                 DCB=(RECFM=VB,LRECL=108,BLKSIZE=27998,BUFNO=30)
//IAMOVRID   DD    *
    ACCESS    DD=MYIAMDS,MINBUFNO=32,MAXBUFNO=64,BACKUPCOMPRESSED
    CREATE    DD=MYIAMDS,CRBUFOPT=MCYL,BACKUPCOMPRESSED
/*
//SYSIN      DD    *
   LISTCAT ENT(my.iam.dataset) ALL
   REPRO INFILE(MYIAMDS) OUTFILE(BACKUP)
   IF MAXCC NE 0 THEN CANCEL
   REPRO  INFILE(BACKUP) OUTFILE(MYIAMDS) REUSE
   LISTCAT ENT(my.iam.dataset) ALL
/*
//
```

Figure 71: Example of Reorganization with IDCAMS, without a Delete and Define (EX1081C)

10.81 CONTINUED . . .

EXAMPLE E: IDCAMS REORG INTO A SECOND IAM DATASET

The following example demonstrates a different approach. In this example, one IAM dataset is copied over another IAM dataset with IDCAMS REPRO REUSE, resulting in a reorganized image of the original dataset. This will reduce the reorganization time by eliminating the writing to and reading from the sequential backup file. After the reorganization, renames of the datasets are done. Using this type of procedure will enable a fast reorganization to minimize the time that the data is not available. The original copy of the dataset can be used for subsequent backup or for read only processing, while normal update processing can then be resumed on the receiving dataset.

```
//REORG     EXEC  PGM=IDCAMS,REGION=0M
//SYSPRINT  DD    SYSOUT=*
//IAMINFO   DD    SYSOUT=*
//IAMPRINT  DD    SYSOUT=*
//MYIAMDS   DD    DSN=my.iam.master.dataset,DISP=OLD
//ALTIAMDS  DD    DSN=my.iam.alternat.dataset,DISP=OLD
//IAMOVRID  DD    *
   ACCESS   DD=MYIAMDS,MINBUFNO=32,MAXBUFNO=64,BACKUPCOMPRESSED
   CREATE   DD=ALTIAMDS,CRBUFOPT=MCYL,BACKUPCOMPRESSED
/*
//SYSIN     DD    *
  LISTCAT ENT(my.iam.dataset) ALL
  LISTCAT ENT(my.iam.alternat.dataset) ALL
  IF MAXCC NE 0 THEN CANCEL
  REPRO INFILE(MYIAMDS) OUTFILE(ALTIAMDS) REUSE
  IF MAXCC NE 0 THEN CANCEL
  ALTER my.iam.dataset NEWNAME(my.iam.tempname.dataset)
  IF MAXCC NE 0 THEN CANCEL
  ALTER my.iam.alternat.dataset NEWNAME(my.iam.dataset)
  IF MAXCC NE 0 THEN CANCEL
  ALTER my.iam.tempname.dataset NEWNAME(my.iam.alternat.dataset)
  LISTCAT ENT(my.iam.dataset) ALL
  LISTCAT ENT(my.iam.alternat.dataset) ALL
/*
//
```

Figure 72: IDCAMS Reorganization to another IAM Dataset (EX1081E)

10.82 DELETING IAM DATASETS

IAM datasets that are not members of an alternate index sphere can be deleted through any manner used to delete non-VSAM datasets, and generally any manner to delete VSAM datasets. This includes through JCL disposition (DISP) processing, various ISPF panels, including the IAM panel, the TSO DELETE command, and the IDCAMS DELETE command.

The IAM DELETE function has been enhanced as part of the alternate index support to perform dataset deletion in a manner that is compatible with VSAM. For example, when deleting a base cluster with alternate index and path associations, all of the associated datasets will be deleted. To use this function, a standard VSAM deletion be performed, rather than a non-VSAM scratch, particularly if the datasets reside on non-SMS managed volumes. Examples of VSAM deletion include the IDCAMS DELETE command, the TSO DELETE command, and the IAM ISPF panel delete function. Examples of functions that use the non-VSAM scratch macro include the ISPF 3.2 and ISPF 3.4 function panels. Performing an ISPF 3.2 deletion of a component of an IAM alternate index sphere will just delete that single dataset, and not the dependent associated datasets.

IDCAMS DELETE

IAM will intercept any IDCAMS DELETE request to determine if the dataset is IAM or VSAM. If the dataset is an IAM dataset, then IAM will issue the system services to perform the delete request. With this feature, existing IDCAMS job steps that issue DELETE commands do not have to be changed when converting datasets to IAM. IAM makes every attempt to follow an appropriate protocol to prevent the recalling of archived or migrated datasets that are being deleted. As with any expiration date protected dataset, the PURGE keyword must be specified if the dataset is so protected. The ERASE parameter on the DELETE CLUSTER is also honored by IAM.

One difference in delete processing from VSAM is that the dataset must not be allocated to any other job or user, otherwise the delete will fail. This is because the deletion or scratch of a non-VSAM dataset internally results in MVS/ESA or OS/390 issuing an exclusive ENQ.

EXAMPLE OF DELETING AN IAM DATASET

In the example below, two IAM datasets are deleted with IDCAMS. The first one has the keyword CLUSTER specified explicitly, so IAM will handle the deletion. The second file is an IAM alternate index dataset. Even though no type is specified, IAM will still process the delete request. In addition to the alternate index, IAM will delete any path datasets that are associated with the alternate index that is being deleted. Both files are allocated in JCL with a DISP=OLD to ensure that this is the only job that is using those datasets. While those DD cards are not necessary, they may prevent deletion failures due to some other job having the dataset(s) allocated.

```
//DELETE    EXEC  PGM=IDCAMS
//SYSPRINT   DD   SYSOUT=*
//FILE1      DD   DSN=my.iam.dataset.file1,DISP=OLD
//FILE2      DD   DSN=my.iam.dataset.aix,DISP=OLD
//SYSIN      DD   *
  DELETE      my.iam.dataset.file1 CLUSTER PURGE
  DELETE      my.iam.dataset.aix PURGE
/*
```

Figure 73: Example of IDCAMS Delete of an IAM Dataset (EX1082A)

10.83 RECURRING IAM DATABASES

RENAMING IAM DATASETS

IAM datasets can be easily renamed by using the IDCAMS ALTER command, the TSO ALTER command, or from the IAM ISPF panels. For Non-SMS managed datasets, the new dataset name remains cataloged in the same catalog as the original dataset name, even if the new high level index is an alias to a different catalog. The NONVSAM version of the ALTER command must be used for IAM datasets.

ALTERNATE INDEX CONSIDERATIONS

IAM datasets that have alternate indexes, are alternate indexes, or are paths have special considerations due to the internally stored association information that is based on the dataset name. The association information is not automatically updated when a rename is performed. Having the correct association information is necessary to prevent failures during OPEN processing, and DELETE processing. To update the association information, it will be necessary to perform an IDCAMS DEFINE RECATALOG on one or more of the associated datasets. IAM has special processing for DEFINE RECATALOG requests to update any dataset associations. The association information can be verified by performing an IDCAMS LISTCAT ALL on the associated datasets. If you are performing renames on more than one of the associated datasets, the DEFINE RECATALOG should be executed after all of the renaming is completed.

The following rules for recataloging must be followed. First, if renaming multiple associated datasets, wait until all the renames are done before performing the recatalogs. Then, the recatalogs can be performed as indicated based on the type(s) of datasets that were renamed. After renaming a base cluster, all of the associated alternate indexes and paths must be recataloged. After renaming an alternate index, the alternate index and any associated paths must be recataloged. After renaming a path, only the path has to be recataloged. The recatalogs must first be performed on the alternate index datasets, and then on the path datasets. Additional information on IDCAMS RECATALOG is in section 10.84.

The IAM ISPF panels have a RENAME function, which will perform the necessary DEFINE RECATALOGS automatically, if consent is given to do so.

EXAMPLE A: RENAME WITH IDCAMS ALTER

This first example demonstrates the use of IDCAMS ALTER to rename an IAM dataset. The high level qualifier is unchanged, so the dataset will remain in the same catalog, unless you are using multilevel alias support.

```
//RENAME    EXEC  PGM=IDCAMS
//SYSPRINT   DD    SYSOUT=*
//IAMPRINT   DD    SYSOUT=*
//SYSIN      DD    *
   LISTCAT  ENT(myiam.origname.dataset) ALL
   ALTER    myiam.origname.dataset       -
               NEWNAME(myiam.newname.dataset)
   LISTCAT  ENT(myiam.newname.dataset) ALL
/*
```

Figure 74: Using IDCAMS ALTER to Rename an IAM Dataset (EX1083A)

10.83 CONTINUED . . .

EXAMPLE B: RENAME INTO DIFFERENT CATALOG

In this next example, an IDCAMS ALTER is again used to change the dataset name. However, in this case the dataset is a non-SMS managed dataset, and the new high level index is an alias in a different catalog than the original high level index. This problem is resolved by doing a DEFINE RECATALOG with the new dataset name, followed by deleting the new entry out of the old catalog.

```
//RENAME    EXEC  PGM=IDCAMS
//SYSPRINT  DD    SYSOUT=*
//IAMPRINT  DD    SYSOUT=*
//SYSIN     DD    *
  LISTCAT ENT(myiam.origname.dataset) ALL
  ALTER myiam.origname.dataset -
      NEWNAME(test.newname.dataset)
  DEFINE CLUSTER(NAME(test.newname.dataset)      -
            OWNER($IAM)                          -
            VOL(volser)                          -
            RECATALOG )                          -
      CATALOG(test.catname)
  DELETE test.newname.dataset NOSCRATCH NONVSAM -
      CATALOG(myiam.catname)
  LISTCAT ENT(test.newname.dataset) ALL
/*
```

Figure 75: Rename into a different catalog (EX1083B)

EXAMPLE C: RENAMING AN ALTERNATE INDEX

In this example, a base cluster, the associated alternate index, and the associated path are all renamed using the IDCAMS ALTER command. Subsequent to the rename, DEFINE REATALOG is issued on the alternate index and the path to update the association information with the new names.

```
//RENAME     EXEC PGM=IDCAMS
//SYSPRINT    DD  SYSOUT=*
//SYSIN       DD  *
       ALTER example.myiam.cluster                   -
             NEWNAME(example.testiam.cluster)
       ALTER example.myiam.aix                       -
             NEWNAME(example.testiam.aix)
       ALTER example.myiam.path                      -
             NEWNAME(example.testiam.path)
       DEFINE ALTERNATEINDEX                         -
             (NAME(example.testiam.aix)              -
       RELATE(example.testiam.cluster)               -
             VOLUME(iamvol)                          -
       RECATALOG )
       DEFINE PATH                                   -
             (NAME(example.testiam.path)             -
             PATHENTRY(example.testiam.aix)          -
             RECATALOG )
       LISTCAT ENT(example.testiam.cluster) ALL
       LISTCAT ENT(example.testiam.aix)     ALL
       LISTCAT ENT(example.testiam.path)    ALL
/*
```

Figure 76: Example of Rename an Alternate Index (EX1083C)

10.83 CONTINUED . . .

EXAMPLE D: RENAME USING IAM ISPF PANELS

The IAM ISPF panels can also be used to rename IAM datasets, and will automatically perform the necessary recatalogs. The rename function is used by selecting option R on the primary IAM panels, and indicating the name of the dataset to be renamed, along with the new name. An example of a completed rename panel is shown below, which will rename the dataset MYIAM.CLUSTER to TESTIAM.CLUSTER.

```
--------------------------- IAM PRIMARY OPTION MENU --------------------------
OPTION ===> r

    I      - Allocate (DEFINE) a new IAM Dataset                     Ver 8.0/01P
    V      - Allocate (DEFINE) a new VSAM Cluster
    D      - Delete a Dataset, Cluster, Path, or Alternate Index
    C      - Copy an IAM Dataset, VSAM KSDS, or VSAM ESDS
    M      - Move an IAM Dataset, VSAM KSDS, or VSAM ESDS
    R      - Rename a Dataset, Cluster, Path, or Alternate Index
    U      - Invoke an IAM Utility program
    blank  - Display IAM Dataset or VSAM Cluster Information


 Enter dataset name (required for all options except U)
 Dataset Name       ===> myiam.cluster
 Dataset Type       ===>    C=Cluster X=AIX P=Path (options I, V Only)

 Enter Model or New dsname (optional for options I and V, required for option R)
 Model|Newname      ===> testiam.cluster


Delete Confirmation ===> YES  Yes|No
```

Figure 77: Example of IAM ISPF Rename

For IAM files that are part of an alternate index association, the following panel will then appear, requesting confirmation to proceed with the recatalog function. Note that in the upper right hand corner the message that the rename is completed. To proceed with the recatalog function, make sure that the Perform Recatalog is set to YES.

```
---------------------------- RECATALOG Components ----------- DATASET RENAMED
COMMAND ===>

   New Dataset Name... RAM2.TESTIAM.CLUSTER

   The above dataset that has been renamed is associated with an IAM
   Alternate Index or Path and needs to be RECATALOGED so IAM can
   maintain the correct relationships with associated datasets.

   Perform RECATALOG ===> YES  (Yes/No)
```

Figure 78:Example of IAM ISPF Recatalog Confirmation Panel

Upon completion of the recatalog, IAM will redisplay the primary IAM panel, with a message in the upper right corner that the recatalog was ok. You can display the renamed dataset and associated components for the association information to verify that the new associations were updated properly.

10.84 RECATALOGING IAM DATASETS

RECATALOGING IAM DATASETS

There are two main reasons for recataloging IAM datasets. The first reason is to rebuild the catalog entry for the dataset, if for some reason it has become uncataloged. The second reason is to update the associated dataset names for datasets that are part of an IAM alternate index sphere, if one or more of the datasets has been renamed, or copied / restored to a new dataset name. The recatalog process is invoked by specifying RECATALOG on the DEFINE CLUSTER, ALTERNATE INDEX, or PATH. In general, only a subset of the operands are required for the recatalog process. The examples of each below will explain the required operands for each type of recatalog.

If you only need to update the catalog entry, an IDCAMS DEFINE NONVSAM request can be used. In this case, if the dataset is SMS managed, then the RECATALOG keyword must be specified, otherwise it should be left off.

EXAMPLE A: RECATALOG OF AN IAM BASE CLUSTER DATASET

In the first recatalog example, an IAM base cluster dataset is being recataloged. For this circumstance, specify the DEFINE CLUSTER command, with the following subparameters:

- Name
- OWNER($IAM), unless $IAM is in the dataset name.
- Volumes on which the dataset resides,
- and RECATALOG.

```
//RECATLOG EXEC  PGM=IDCAMS
//SYSPRINT  DD   SYSOUT=*
//IAMPRINT  DD   SYSOUT=*
//SYSIN     DD   *
      DEFINE CLUSTER(NAME(my.notcat.iam.dataset)       -
             OWNER($IAM)                               -
             VOLUMES(iam001 iam002)                    -
             RECATALOG)
      LISTCAT ENT(my.notcat.iam.dataset) ALL
/*
```

Figure 79: Recatalog an IAM Dataset (EX1084A)

10.84 CONTINUED . . .

EXAMPLE B: RECATALOG USING DEFINE NONVSAM

In this example, an SMS managed IAM dataset is recataloged using a DEFINE NONVSAM. The following subparameters are used for this request:

- **NAME –** identifies the dataset name.
- **VOLUMES –** identifies each volume that the dataset resides on.
- **DEVT –** for each volume in the above list, a device type must be specified.
- **OWNER($IAM) –** The OWNER parameter is optional on the DEFINE NONVSAM, but is recommended so that the catalog entry will resemble the one that IAM builds.
- **RECATALOG –** This keyword must be provided if the dataset is SMS managed, otherwise it must not be specified.

```
//RECATSMS EXEC  PGM=IDCAMS
//SYSPRINT  DD   SYSOUT=*
//IAMPRINT  DD   SYSOUT=*
//SYSIN     DD   *
      DEFINE NONVSAM(NAME(my.notcat.iam.dataset)     -
           OWNER($IAM)                                -
           VOLUMES(iam001 iam002)                     -
           DEVT(3390 3390)                            -
           RECATALOG)
      LISTCAT ENT(my.notcat.iam.dataset) ALL
/*
```

Figure 80: Recatalog Using DEFINE NONVSAM (EX1084B)

EXAMPLE C: RECATALOG AN IAM ALTERNATE INDEX

In this example, an IAM Alternate Index is being recataloged. The operation is quite similar to the recatalog of a base cluster, but with a few differences in the operands. The related base cluster must be specified, and the OWNER($IAM) operand is completely optional. When recataloging an alternate index, the association with the base cluster information is also updated both in the IAM alternate index dataset, and also in the IAM base cluster. The following operands are used:

- **NAME –** Indicates the name of the IAM Alternate Index.
- **RELATE –** Indicates the name of the base cluster. If the base cluster has been renamed, use the new name here. The base cluster must be cataloged prior to recataloging the alternate index.
- **VOLUMES –** Identifies the volume(s) on which the alternate index resides.
- **RECATALOG –** Indicates that the dataset is already existing, and that the catalog entry is just being rebuilt, along with updating the base cluster association information.

```
//RECATAIX EXEC PGM=IDCAMS
//SYSPRINT  DD  SYSOUT=*
//SYSIN     DD  *
      DEFINE ALTERNATEINDEX                          -
            (NAME(example.testiam.aix)               -
            RELATE(example.testiam.cluster)          -
            VOLUME(myvol1)                           -
            RECATALOG )
      LISTCAT ENT(RAM2.TESTIAM.AIX)     ALL
/*
```

Figure 81: Example of Recataloging an IAM Alternate Index (EX1084C)

10.84 CONTINUED . . .

EXAMPLE D: RECATALOGING AN IAM PATH

Because an IAM PATH is an actual dataset, rather than just a catalog entry, the recatalog of an IAM PATH does have a special consideration, in that the PATH must be cataloged for the RECATALOG to work. This is because IDCAMS provides no mechanism to pass a volume on the DEFINE PATH command. So, if the PATH is not cataloged, it must be preceded by a DEFINE NONVSAM of the PATH, followed by the DEFINE RECATALOG. In this case, the need for the RECATALOG is to update the association information in the path itself, and the path entry dataset. The example below demonstrates how this is done. First, a LISTCAT is done on the path to see if it is cataloged. If it is not cataloged, then a DEFINE NONVSAM is issued to catalog the path. Then the DEFINE PATH RECATALOG is done.

```
//RECATPTH EXEC  PGM=IDCAMS
//SYSPRINT  DD   SYSOUT=*
//SYSIN     DD   *
      LISTC ENT(RAM2.TESTIAM.PATH)
      IF LASTCC NE 0 THEN DO
            SET   MAXCC=0
            DEFINE NONVSAM                      -
                  (NAME(RAM2.TESTIAM.PATH)      -
                  VOLUMES(JUNK01)               -
                  DEVT(3380)                    -
                  OWNER($IAM) )
            END
            ELSE
      DEFINE PATH                          -
            (NAME(RAM2.TESTIAM.PATH)       -
            PATHENTRY(RAM2.TESTIAM.AIX)    -
            RECATALOG )
      LISTCAT ENT(RAM2.TESTIAM.PATH)     ALL
/*
```

Figure 82: Example of Recatalog an IAM Path (EX1084D)

10.85 BACKING UP AND RESTORING IAM DATASETS

BACKUP OVERVIEW

Frequently, IAM datasets are incorporated into normal DASD management procedures for backup, restore, and archival (or migration) purposes. Software products that provide such services can easily handle IAM datasets within the normal functionality that they provide. The major considerations are first, that IAM datasets are treated as non-VSAM datasets by these products, and second that the default DSORG of PS for IAM files is highly recommended to provide the most flexibility when managing IAM datasets with such products. The functionality and limitations that these products have with non-VSAM datasets will apply for IAM datasets.

Many application job streams incorporate their own backup up of their related datasets in scheduled time frames that are more relevant to the application, and provide for improved recoverability of application data. As an aid to those responsible for maintaining recoverability of application data contained in IAM datasets, this section will present some of the considerations with the various methods of backing up and restoring at the dataset level, along with some examples. The examples are intended to demonstrate basic functionality of the various backup and restore methods that are applicable to IAM datasets.

ALTERNATE INDEX CONSIDERATIONS

For IAM datasets in an associated alternate index sphere, as with VSAM, the most critical piece is the base cluster dataset. After restoring a base cluster, the alternate indexes and paths can always be recreated if backups are not readily available. The major concern with IAM alternate indexes and their associated datasets is that DASD management utility programs will treat each dataset as a separate entity. With VSAM, some of these utilities have developed processing to recognize the associations, and handle the base cluster and all associated alternate indexes and paths as one entity. Using dataset naming conventions within a set of alternate index associated datasets where the only name difference is in the last index level of the name will help ease the dataset management difficulties. By doing so, you can treat any set of associated datasets by a dataset group name in the control cards for the utility. Information on how to do this will be provided in some of the subsequent examples.

BACKUP WITH IDCAMS

The most common method for backing up IAM base cluster and alternate index datasets by application job streams is with IDCAMS REPRO. This method copies the data at the record level into a sequential output dataset. The main advantages of using such a backup are that application programs can directly read the data from the backup sequential dataset, a restore will reorganize the dataset, and when moved to a different device type, IAM will adjust the block size automatically to obtain the best possible device utilization. Performance of such a backup can be very fast through effective use of buffering. If the sequential copy of the data is only used for backup purposes, then the IAM Backup Compressed feature can be used. This feature prevents the decompressing of the data during backup, and the compressing of data when it is used to reload the dataset. Using the IAM Backup Compressed feature saves CPU time, as well as I/O time transferring data to and from the backup media. There is unlikely to be any savings in terms of the number of tapes required to backup the IAM dataset when the tape devices that also offer compression, because the data is already compressed. The main disadvantage of IDCAMS REPRO is that the attributes used to define the dataset are not retained with the output dataset. While IDCAMS does offer a function for VSAM datasets to save the dataset attributes, called EXPORT, and the corresponding restoration of dataset and attributes with IMPORT, IAM does not provide support for the use of those functions.

BACKUP AND RESTORE UTILITIES

An alternative to using IDCAMS is to use a software product that provides DASD dataset management functions, including backing up and restoring datasets. Examples of such products include FDR, ABR, DFSMSdss, and DFSMShsm. These products offer the fastest possible data movement, along with various dataset management capabilities. A major advantage of using one of these products is that the file attribute information is saved with the backup, along with an exact image of the dataset. The disadvantages are that application programs can not directly access the data because of the format it is stored in, the software is not as easy to use as an IDCAMS REPRO, when restored, the dataset is not reorganized, and restoration to an unlike device can cause inefficient usage of DASD space. However, when fast backup times are required, and when the backup copy of the data is not being used by the application, then these DASD utilities will meet that requirement.

10.85 CONTINUED . . .

OTHER BACKUP OPTIONS

There are other utility software products that also provide a data movement capability for VSAM datasets that can also be used for IAM datasets. Examples include using the DFSORT, SYNCSORT, or using DITTO. These and similar software products essentially work like an IDCAMS REPRO, in that they process the data at the logical record level using standard access method I/O requests. Software packages that use control interval (CI) access to process VSAM datasets or do their own I/O rather than using VSAM, such as FAVER or BMC VSAM-ASSIST, will not work with IAM datasets.

USING IDCAMS REPRO

The use of IDCAMS REPRO for backing up and restoring IAM datasets is straight forward. Jobs that have been set up to backup VSAM datasets with this method should not require any changes. One thing to be careful of is that IDCAMS defaults sequential output files to RECFM=U, which forces each output record to become a physical block on the storage media. This can waste media and backup time. By assigning the output dataset a RECFM=VB, and a block size of 27998 for 3390 DASD, or 32760 for tape, plus providing a large number of buffers, will speed up the REPRO process substantially. When an installation uses the IAM buffering defaults as shipped, then there should be no need for IAM overrides on an IDCAMS repro, unless you desire to use BACKUPCOMPRESSED.

EXAMPLE A: IDCAMS BACKUP

In this example, a large multi-volume IAM dataset is backed up by an IDCAMS REPRO. The output media is tape. Note that the DCB information that is provided including BUFNO. The LRECL for the output tape volume is the maximum record size defined for the IAM dataset plus four for the RDW added for variable format records and an additional four for IAM compression data. So, assuming the IAM file has a maximum record length of 1016, a value of at least 1024 must be used for the LRECL on the output dataset. The BACKUPCOMPRESSED override is specified because the sequential output file will not be used for any purpose other than to reload the dataset should a restore be needed. A volume count of 20 is provided for the output tape dataset to provide enough tape volumes to hold the data. A LISTCAT ALL is included so that information on what is needed to define the dataset, if necessary, is readily available.

```
//BACKUP    EXEC  PGM=IDCAMS
//SYSPRINT  DD    SYSOUT=*
//IAMINFO   DD    SYSOUT=*
//IAMPRINT  DD    SYSOUT=*
//BACKUP    DD    DSN=my.iam.dataset.backup,DISP=(,CATLG),
//     DCB=(RECFM=VB,LRECL=1024,BLKSIZE=32760,BUFNO=30),
//     UNIT=3490,VOL=(,,,20)
//IAMFILE   DD    DSN=my.iam.dataset,DISP=OLD
//IAMOVRID  DD    *
   ACCESS   DD=&ALLDD,BACKUPCOMPRESSED
/*
//SYSIN     DD    *
  LISTCAT ENT(my.iam.dataset) ALL
  REPRO INFILE(IAMFILE) OUTFILE(BACKUP)
/*
```

Figure 83: Example of IDCAMS Backup (EX1085A)

10.85 CONTINUED . . .

EXAMPLE B: IDCAMS RESTORE

The following example demonstrates how to restore the dataset that was backed up in the prior example. The IAM dataset is not being deleted and redefined, so the REUSE parameter is required on the REPRO statement. A BUFNO is provided on the tape input to speed up reading the data. A CREATE override indicating BACKUPCOMPRESSED is required because the backup tape contains IAM compressed data. A LISTCAT ALL is done after the REPRO to verify the new file structure.

```
//RESTORE   EXEC  PGM=IDCAMS
//SYSPRINT  DD    SYSOUT=*
//IAMPRINT  DD    SYSOUT=*
//IAMINFO   DD    SYSOUT=*
//BACKUP    DD    DSN=my.iam.dataset.backup,DISP=OLD,
//     DCB=BUFNO=30,UNIT=3490
/IAMFILE    DD    DSN=my.iam.dataset,DISP=OLD
//IAMOVRID  DD    *
   CREATE   DD=&ALLDD,BACKUPCOMPRESSED
/*
//SYSIN     DD    *
  REPRO INFILE(BACKUP) OUTFILE(IAMFILE) REUSE
  LISTCAT ENT(my.iam.dataset) ALL
/*
```

Figure 84: Example of an IDCAMS Restore with BACKUPCOMPRESSED (EX1085B)

EXAMPLE C: DECOMPRESS THE BACKUP DATASET

If ever needed, the IAMRECVR utility program can be used to create a sequential uncompressed file from the sequential file with records in an IAM data compressed format, as created by using the BACKUPCOMPRESSED override. The new file will contain uncompressed records from the original dataset, ready to be used by application or other processing needs. To perform this function, you must know the key length and key offset (RKP) for the file being processed. These values can be obtained from a LISTCAT or an IAMINFO report from the original dataset. In the example below, a new sequential backup with uncompressed records is created using the sequential dataset created in Example 1. The DISKIN DD specifies the original backup dataset, and the TAPEOUT DD specifies the new uncompressed dataset.

```
//DECOMPRS EXEC PGM=IAMRECVR,REGION=0M
//SYSPRINT  DD  SYSOUT=*
//DISKIN    DD  DSN=my.iam.dataset.backup,DISP=OLD
//TAPEOUT   DD  DSN=my.iam.dataset.uncomp,UNIT=TAPE,
//              DISP=(,CATLG),
//              DCB=(RECFM=VB,LRECL=1020,BLKSIZE=32760)
//SYSIN     DD  *
  DECOMPRESS    KEYLEN=16,RKP=0
/*
```

Figure 85: Example of decompressing a BACKUPCOMPRESSED Dataset (EX1085C).

10.85 CONTINUED . . .

SINGLE VOLUME

For the DASD backup and restore utilities, the examples will be split into single volume datasets, and multivolume datasets. Single volume IAM datasets are very easily dumped and restored with either FDR or DFSMSdss. While normally not necessary, it is recommended that the original dataset be deleted prior to doing the restore if you are replacing the original dataset. When the delete is done, the restore will allocate the dataset attempting to obtain all the required space in one extent, that may be beneficial because the file may be able to acquire more DASD space.

The single volume examples shown below are all set to handle either a single IAM dataset, or all associated IAM datasets in an alternate index sphere, if they are all on the same volume. If the various associated datasets are on different volumes, then use the multivolume procedures. For these examples, each associated dataset in the alternate index sphere has an identical high level quailifier(s), with only the last level being different. Assume the following names:

- Base Cluster DSN=my.iam.dataset.cluster
- Alternate Index DSN=my.iam.dataset.aix
- Path DSN=my.iam.dataset.path

If you are using different naming conventions, refer to the manual for the product you use for further information on building a dataset name mask.

EXAMPLE D: FDR/DSF BACKUP

In this first example, FDR is being used to backup an IAM dataset to tape. The JCL requirements are for a DISKx DD statement that specifies the volume of the dataset being backed up. Then a corresponding TAPEx DD statement is required for the tape to contain the backup data.

```
//BACKUP    EXEC PGM=FDRDSF,REGION=0M
//SYSPRINT DD   SYSOUT=*
//DISK1     DD   DSN=my.iam.dataset.cluster,DISP=OLD
//TAPE1     DD   DSN=my.iam.dataset.backup,DISP=(,CATLG),UNIT=TAPE
//SYSIN     DD   *
 DUMP TYPE=DSF
 S DSN=my.iam.dataset.**
/*
```

Figure 86: Example of FDR/DSF Backup of Single Volume IAM Dataset (EX1085D)

EXAMPLE E: FDR/DSF RESTORE

The restore of the dataset is just as easy. This example works whether or not the original dataset had been deleted. The DISK1 DD specifies the receiving volume, which does not have to be the same as the dataset originally resided on. The TAPE1 DD specifies the backup tape created in the prior backup example. The backup tape could also be from a full volume backup as well.

```
//RESTORE   EXEC PGM=FDRDSF,REGION=0M
//SYSPRINT DD   SYSOUT=*
//DISK1     DD   UNIT=3390,VOL=SER=iamvol,DISP=OLD
//TAPE1     DD   DSN=my.iam.dataset.backup,DISP=OLD
//SYSIN     DD   *
 RESTORE TYPE=DSF
 S DSN=my.iam.dataset.**
/*
```

Figure 87: Example of an FDR/DSF restore of a Single Volume IAM Dataset (EX1085E)

10.85 CONTINUED . . .

EXAMPLE F: DFSMSDSS BACKUP

In this next example, DFSMSdss is being used to backup an IAM dataset. The output tape is specified by the OUTDD parameter, and use of an input DD is optional, and not shown in this example.

```
//BACKUP    EXEC PGM=ADRDSSU,REGION=0M
//SYSPRINT DD   SYSOUT=*
//TAPE1    DD   DSN=my.iam.dataset.backup,DISP=(,CATLG),UNIT=TAPE
//SYSIN    DD   *
  DUMP OUTDD(TAPE1) DS(INCL(my.iam.dataset.**))
/*
```

Figure 88: Example of DFSMSdss Dump of a Single Volume IAM Dataset (EX1085F)

EXAMPLE G: DFSMSDSS RESTORE

This corresponding example demonstrates how to restore the IAM dataset. The output volume could be the same as the original, or different.

```
//RESTORE   EXEC PGM=ADRDSSU,REGION=0M
//SYSPRINT DD   SYSOUT=*
//DISK1    DD   UNIT=3390,VOL=iamvol,DISP=OLD
//TAPE1    DD   DSN=my.iam.dataset.backup,DISP=OLD
//SYSIN    DD   *
  RESTORE DS(INCL(my.iam.dataset.**)) -
    INDD(TAPE1) OUTDD(DISK1) REPLACE CATALOG
/*
```

Figure 89: Example of DFSMSdss Restore of a Single Volume IAM Dataset (EX1085G)

MULTIVOLUME BACKUP & RESTORE

Multivolume backup and restore operations require a little more effort. The best overall process for the combined backup and restore effort seems to be using the application backup capabilities of ABR. While this does require an additional job step on both the backup and restore jobs, the dataset is restored across the same number of volumes as it had originally, along with the same amount of space on each volume. If the ABR product is not available, then the same results can still be achieved with FDR/DSF with a little more effort on the JCL. DFSMSdss can also be used for multivolume backup and restore, although it may be difficult to restore the dataset spread across multiple volumes as it was originally.

The multivolume backup and restore process can also be used when backing up an IAM alternate index sphere, where the datasets reside on different volumes, or one or more of the datasets resides on multiple volumes. These set of examples will continue to assume use of the dataset naming conventions from the single volume examples.

10.85 CONTINUED . . .

EXAMPLE H: ABR BACKUP

In this example, a multivolume IAM dataset will be backed up using ABR application backup and recovery capability. While the example given here is just a single dataset, multiple datasets for the entire application can be backed up and restored as necessary with this technique. Basically, this process involves creating a temporary archive control file, which is used to store information about the dataset(s) backed up, and is subsequently also backed up to the same tape as the dataset(s). If this type of backup job is being set up to run on a regular basis, then GDG dataset names can be used for both the Archive control file, and the backup of the Archive control file. Note that ABR requires that the word ARCHIVE be included within the dataset name for the Archive control file. The name of the backup of the Archive control file will be similar, only with ARCBKUP in place of ARCHIVE. The Archive control file can be deleted after this job runs. Full details on using this capability of ABR can be found in section 52 of the FDR manual.

```
//INITARCH EXEC PGM=FDRARCH
//SYSPRINT DD   SYSOUT=*
//ARCHIVE  DD   DSN=my.ARCHIVE.dataset,UNIT=SYSDA,DISP=(,CATLG),
//    SPACE=(TRK,(5,5),RLSE)
//SYSIN    DD   *
 FORMAT RECS=nnnn,USERINDEX=YES  flset nnnn to number of datasets
/*
//BACKUP   EXEC PGM=FDRABR,REGION=0M
//SYSPRINT DD   SYSOUT=*
//SYSPRINA DD   SYSOUT=*
//ARCHIVE  DD   DSN=my.ARCHIVE.dataset,DISP=OLD
//TAPEA    DD   DSN=my.backup.dataset,UNIT=TAPE,DISP=(,KEEP)
//SYSIN    DD   *
  DUMP     TYPE=APPL,ONLVOL,RTC=YES,
           ARCBACKUP=DSF,DSNENQ=HAVE
  SELECT   CATDSN=my.iam.dataset.**
/*
```

Figure 90: Example of FDR/ABR Backup of Multivolume IAM Dataset (EX1085H)

10.85 CONTINUED . . .

EXAMPLE I: ABR RESTORE

The restore process is a two step process as well. The first step restores the copy of the archive control file that was backed up, then the next step restores the IAM dataset. In this example the dataset is being restored to the original volumes, however there are various options to change the volumes that it is being restored to, as well as to give the dataset a new name. When restoring as the original dataset, it is highly recommended that the dataset be deleted first, so that ABR can allocate it as it existed previously.

```
//DELETE    EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  DELETE my.iam.dataset.cluster
  DELETE my.iam.dataset.cluster NOSCRATCH
  SET MAXCC=0
/*
//RESTARC   EXEC PGM=FDRABR,REGION=0M
//SYSPRINT DD   SYSOUT=*
//SYSPRINA DD   SYSOUT=*
//TAPEA    DD   DSN=my.ARCBKUP.dataset,VOL=(,RETAIN),DISP=OLD
//SYSIN    DD   *
  RESTORE   TYPE=ABR,RECAT
  SELECT    DSG=my.ARCHIVE,TAPEDD=A
/*
//RESTDATA EXEC PGM=FDRABR,REGION=0M
//SYSPRINT DD   SYSOUT=*
//ARCHIVE  DD   DSN=my.ARCHIVE.dataset,DISP=OLD
//TAPE1    DD   DSN=my.FDR,DISP=OLD,VOL=REF=*.RESTARC.TAPEA
//SYSIN    DD   *
  RESTORE TYPE=APPL,RECAT
  SELECT   DSN=my.iam.dataset.**
/*
```

Figure 91: Example of FDR/ABR Restore of Multivolume IAM Dataset (EX1085I)

EXAMPLE J: FDR/DSF MULTIVOLUME BACKUP

FDR/DSF can also be used to backup and restore a multivolume IAM dataset, although the use of ABR Application backup is strongly recommended. A three volume IAM dataset is being backed up in this example. Each volume is effectively a separate backup, but can and should be done in one job step. For each DASD volume, there will be a DISKx and corresponding TAPEx DD card. As long as you make sure that you get all the pieces on both the backup and restore, this process will work fine.

```
//BACKUP    EXEC PGM=FDRDSF,REGION=0M
//SYSPRINT DD   SYSOUT=*
//DISK1    DD   UNIT=3390,DISP=OLD,VOL=SER=iam001
//DISK2    DD   UNIT=3390,DISP=OLD,VOL=SER=iam002
//DISK3    DD   UNIT=3390,DISP=OLD,VOL=SER=iam003
//TAPE1    DD   DSN=my.iam.backup.vol1,UNIT=TAPE,DISP=(,CATLG)
//TAPE2    DD   DSN=my.iam.backup.vol2,UNIT=TAPE,DISP=(,CATLG)
//TAPE3    DD   DSN=my.iam.backup.vol3,UNIT=TAPE,DISP=(,CATLG)
//SYSIN    DD   *
  DUMP      TYPE=DSF
  SELECT    DSN=my.iam.dataset.**
/*
```

Figure 92: Example of using FDR/DSF to Backup a Multivolume IAM Dataset (EX1085J)

10.85 CONTINUED . . .

EXAMPLE K: FDR/DSF MULTIVOLUME RESTORE

To restore the dataset, the procedure is almost identical. In this example, the dataset is being restored to the same three volumes, however different volumes could be used, and the dataset could be given a new name. If the dataset is being restored to replace the existing version, it is highly recommended that the dataset be deleted prior to performing the restore. To insure that the dataset does not exist, the restore is preceded by an IDCAMS step to delete the dataset.

```
//DELETE    EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  DELETE my.iam.dataset.cluster
  DELETE my.iam.dataset.cluster NOSCRATCH
  SET MAXCC=0
/*
//RESTORE  EXEC PGM=FDRDSF,REGION=0M
//SYSPRINT DD   SYSOUT=*
//DISK1    DD   UNIT=3390,DISP=OLD,VOL=SER=iam001
//DISK2    DD   UNIT=3390,DISP=OLD,VOL=SER=iam002
//DISK3    DD   UNIT=3390,DISP=OLD,VOL=SER=iam003
//TAPE1    DD   DSN=my.iam.backup.vol1,DISP=OLD
//TAPE2    DD   DSN=my.iam.backup.vol2,DISP=OLD
//TAPE3    DD   DSN=my.iam.backup.vol3,DISP=OLD
//SYSIN    DD   *
  RESTORE  TYPE=DSF
  SELECT   DSN=my.iam.dataset.**
/*
```

Figure 93: Example of Using FDR/DSF to Restore a Multivolume IAM Dataset (EX1085K)

EXAMPLE L: DFSMSDSS MULTIVOLUME BACKUP

In this last pair of examples, DFSMSdss is being used to backup and subsequently restore a multivolume IAM dataset. As can be seen, the backup portion of this example is essentially identical to the backup of a single volume IAM dataset.

```
//BACKUP    EXEC PGM=ADRDSSU,REGION=0M
//SYSPRINT DD   SYSOUT=*
//TAPE1    DD   DSN=my.iam.backup,UNIT=TAPE,DISP=(,CATLG)
//SYSIN    DD   *
  DUMP     OUTDD(TAPE1) DS(INCL(my.iam.dataset.**))
/*
```

Figure 94: Example of DFSMSdss Backup of a Multivolume IAM Dataset (EX1085L)

10.85 CONTINUED . . .

EXAMPLE M: DFSMSDSS MULTIVOLUME RESTORE

Now, in this example DFSMSdss will be used to restore the dataset backed up in the prior example. Unless you can be certain that the IAM dataset was not deleted and defined or reorganized since the backup was done, the dataset should be deleted prior to attempting the restore. This is to prevent a possible restore failure. To force the dataset to be split across the three volumes, temporary datasets are allocated on all of the DASD volumes so that DFSMSdss will not find enough space to restore the whole dataset on any particular volume. The manual part of this process involves checking each volume for how much space it has available, and determining how much space is needed by the dataset being restored, and how much space should be used on each volume for the dataset being restored. Then, set up the JCL to allocate a temporary dataset on each volume with the remaining free space.

```
//DELETE    EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  DELETE my.iam.dataset.cluster
  DELETE my.iam.dataset.cluster NOSCRATCH
  SET MAXCC=0
/*
//RESTORE  EXEC PGM=ADRDSSU,REGION=0M
//SYSPRINT DD   SYSOUT=*
//TEMP1    DD   UNIT=3390,VOL=SER=iam001,SPACE=(CYL,2000)
//TEMP2    DD   UNIT=3390,VOL=SER=iam002,SPACE=(CYL,1500)
//TEMP3    DD   UNIT=3390,VOL=SER=iam003,SPACE=(CYL,400)
//TAPE1    DD   DSN=my.iam.backup,DISP=OLD
//SYSIN    DD   *
  RESTORE  DATASET(INCLUDE(my.iam.dataset.**)) -
      INDD(TAPE1) ODY((iam001),(iam002),(iam003)) -
      REPLACE CATALOG
/*
```

Figure 95: Example of DFSMSdss Restore of a Multivolume IAM Dataset (EX1085M)

10.86 MOVING IAM DATASETS

MOVING IAM DATASETS

There are several different software utilities that can be used to move IAM datasets. When IAM datasets are being moved along with many other datasets, for example as part of a reconfiguration, DASD backup and restore software, such as FDR from Innovation or DFSMSdss from IBM are frequently used. Other related options are to move the dataset(s) with FDRCOPY, or with DFSMShsm. These utilities will accomplish the dataset movement easily, providing that the DSORG of the IAM datasets is PS. The problem with using these products, particularly when moving IAM datasets to devices with a different geometry, are that the moved dataset(s) may not make as effective or efficient use of DASD space as is possible. For example, moving an IAM dataset that is blocked at the typical _ track block size on a 3380, to a 3390 will result in not using 16% of the track capacity. Even when moving to the same device type, the datasets will not be reorganized resulting in the movement of unused areas within the dataset.

MOVING WITH FDRREORG

The best way from an internal dataset structure and DASD space view point, is to move IAM datasets by performing a dataset reorganization. FDRREORG provides an excellent tool for moving IAM dataset(s) to a different volume(s). FDRREORG not only knows the basic file attributes, it also knows what IAM Overrides were used to create the original dataset, and will pass those override values on the define of the new dataset. The other major alternative is to use IDCAMS REPRO, although that does involve more manual effort than FDRREORG.

EXAMPLE A: USING FDRREORG TO MOVE IAM DATASETS

Shown below is an example of moving all of the IAM datasets from one volume to another. The NODEFAULTS keyword means that the selection criteria is based entirely on the control card input. The NOUPDATES=YES causes the datasets to be reorganized, even if there were no updates, inserts, or deletes. The IAMDEFINE=YES forces the delete and define of the IAM datasets. On the Select card, all IAM datasets are selected from one volume, and moved to another volume. The NEWALL keyword is required so that the new volume will be applied to all of the selected datasets.

```
//IAMDSMOV EXEC PGM=FDRREORG,REGION=0M
//SYSPRINT DD   SYSOUT=*
//REORGPRT DD   SYSOUT=*
//REORGRPT DD   SYSOUT=*
//IAMINFO  DD   SYSOUT=*
//SYSIN    DD   *
  REORG    NODEFAULTS,NOUPDATES=YES,IAMDEFINE=YES
  SELECT   ALLDSN,DSTYPE=IAM,VOL=oldvol,
            NEWVOLSDATA=newvol,NEWALL
/*
```

Figure 96: Example of Using FDRREORG to Move IAM Datasets (EX1086A)

10.86 CONTINUED . . .

EXAMPLE B: USING FDREPORT TO GENERATE IDCAMS CONTROL CARDS

Another alternative automation technique for moving of IAM datasets by reorganization is to use FDREPORT. As demonstrated below, FDREPORT can be used to create a dataset containing control cards for IDCAMS. The following functions will be performed by the IDCAMS control cards:

1. Define an IAM dataset on the new volume, modeled from the original dataset.
2. Copy the original dataset into the new dataset.
3. Delete the original dataset.
4. Rename the new dataset to the original dataset name.
5. Do a LISTCAT on the moved dataset.

If an error occurs, then the IDCAMS execution is stopped. The model control cards for each dataset is specified by the data in the MASK DD input stream. By control card specifications to FDREPORT, only IAM datasets are selected that are single volume datasets cataloged to the volume that has been selected. One could get more sophisticated with the IF logic control cards created for IDCAMS, so that processing will resume with the next dataset rather than stopping completely on the first error.

```
//SELECTDS EXEC PGM=FDREPORT,REGION=0M
//SYSPRINT  DD  SYSOUT=*
//SYSPUNCH  DD  DSN=&&CAMS,UNIT=SYSDA,SPACE=(CYL,(2,1)),
//              DISP=(,PASS),
//              DCB=(RECFM=FB,LRECL=80,BLKSIZE=23440)
//SYSIN     DD  *
 XSELECT DSORG=IAM,VOL=oldvol,CATALOG=YES,CATVOLCT=1
 PUNCH FDRLIB=MASK
 PRINT ENABLE=IAM,RPTYPE=SELPCH,SORT=NO
/*
//MASK      DD  *
 DEFINE CLUSTER(NAME(<DSN>.NEW) -
      MODEL(<DSN>)     -
      OWNER($IAM)      -
      VOL(newvol)   )
  IF MAXCC NE 0 THEN CANCEL
 REPRO IDS(<DSN>) ODS(<DSN>.NEW)
  IF MAXCC NE 0 THEN CANCEL
 DELETE <DSN>
 ALTER <DSN>.NEW NEWNAME(<DSN>)
  IF MAXCC NE 0 THEN CANCEL
 LISTCAT ENT(<DSN>) ALL
)SUFFIX
  IF MAXCC NE 0 THEN CANCEL
/*
//MOVEIAM EXEC PGM=IDCAMS,COND=(0,NE)
//SYSPRINT DD  SYSOUT=*
//IAMPRINT DD  SYSOUT=*
//IAMINFO  DD  SYSOUT=*
//SYSIN    DD  DSN=&&CAMS,DISP=OLD
```

Figure 97: Example of using FDREPORT to Generate Control Cards for IDCAMS (EX1086B)

10.86 CONTINUED . . .

EXAMPLE C: MOVING DATASETS WITH FDRCOPY

FDRCOPY can also be used to move IAM datasets. The advantage of using FDRCOPY is that all or a selected subset of datasets can be copied, regardless of type, in one job step. The disadvantage, as discussed above, is that IAM datasets will not be reorganized or reblocked. However, it is quite easy to use. In the following example all non-VSAM datasets are being moved, which will include any IAM datasets.

```
//MOVENVSM EXEC PGM=FDRCOPY,REGION=0M
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  MOVE     TYPE=DSF
  SELECT   ALLDSN,VOL=oldvol,NVOL=newvol
/*
```

Figure 98: Example of Moving Datasets with FDRCOPY (EX1086C)

EXAMPLE D: MOVING AN IAM DATASET WITH FDRCOPY

You can also use FDRCOPY to move a single or group of IAM datasets. In the example below, a single IAM dataset is moved with FDRCOPY to a new volume, as identified by the NVOL parameter.

```
//MOVEDS   EXEC PGM=FDRCOPY,REGION=0M
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   *
  MOVE     TYPE=DSF
  SELECT   CATDSN=my.iam.dataset,,NVOL=newvol
/*
```

Figure 99: Example of Movind Datasets with FDRCOPY (EX1086D)

EXAMPLE E: MOVING DATASETS WITH DFSMSDSS

A similar type of dataset move function can also be accomplished with DFSMSdss. The move operation is effected by specifying the DELETE and RECATALOG(*) keywords on the control card. Non-VSAM datasets with organizations of sequential, partitioned, or direct are being moved. The AUTORELBLKA keyword is specified to insure that IAM datasets with DSORG=DA are copied properly.

```
//KSD972C EXEC PGM=ADRDSSU,REGION=0M,COND=EVEN
//SYSPRINT  DD SYSOUT=*
//SYSIN     DD *
 COPY DATASET (INCLUDE(**)                          -
   BY((DSORG EQ (SAM,PDS,BDAM)) (CATLG EQ YES)) )  -
   LIDY((oldvol)) ODY((newvol))                     -
   DELETE RECATALOG(*) AUTORELBLKA
/*
```

Figure 100: Example of Moving Datasets with DFSMSdss (EX1086E)

10.87 RECOVERING IAM DATASETS

RECOVERY As with any dataset containing critical data, it is important to have established dataset recovery procedures in place. This includes having proper dataset backups and a way to restore any potential data that may be lost due to a hardware or software failure. Most software packages that provide logging and recovery for VSAM files can also be used with IAM datasets. To assist in dataset recovery, IAM provides a utility program, IAMRECVR, that may be of assistance in the recovery process. IAMRECVR is not intended to take the place of established dataset recovery procedures, but may aid in reducing the time required to recover a dataset, or prevent the use of such procedures in some circumstances. IAMRECVR can only retrieve the data that physically exists on the DASD device in a readable form. Data that is inaccessible due to media or other failures are not recoverable with this utility. Likewise, data that has been overwritten or never written out to the storage media can not be recovered by IAMRECVR.

IAM Alternate index datasets can be recovered with either the IAMRECVR utility, or by rebuilding the alternate index using the IDCAMS BLDINDEX function. Using IDCAMS may be preferable, particularly when there is potential data loss, such as may occur with a hardware data check, or other media failure.

IAMRECVR Should you suspect any type of problem with an IAM file, either physical (I/O errors), or software failures, IAMRECVR can be used to aid in diagnosing and recovering IAM datasets from such types of errors. IAMRECVR offers a set of services that includes the following commands:

1. DIAGNOSE – Read the entire dataset to validate general dataset integrity.
2. PRINT – Print selected portions of the IAM dataset to provide diagnostic information.
3. RECOVER – Reads all of the data in the IAM dataset, and copies the readable data to a sequential output dataset that can be used to reload the file. Records with duplicate keys are optionally written to a separate sequential dataset.
4. APPLY – Copies data from the duplicate log file into the newly loaded recovered dataset.

IAMRECVR has knowledge about underlying structure of an IAM dataset, and reads IAM files without using the IAM access method. IAMRECVR utilizes high performance I/O, reading up to an entire cylinder per physical I/O. With the information about each particular dataset, which is usually retrieved from the dataset itself, IAMRECVR pulls out data records from the file. While the entire dataset will be read, data blocks containing the index structure are not processed.

This section will explain how to validate dataset integrity, how to collect any diagnostic information that may be requested, and how to recover the dataset from various error situations through the use of IAMRECVR and various other utilities. Complete information on all of the functions and operands available with IAMRECVR are provided in the Reference Section of the manual. The majority of dataset integrity problems are the result of improper dataset sharing. In those circumstances, there may quite likely be some data lost because it may have been overwritten, and can not be retrieved because it is not there. Complete data recovery will require the use of additional recovery procedures beyond those provided by IAM.

There are two general categories of failures. The first category consists of errors encountered processing the dataset. For example, you may have a program that is receiving an IAMW12 message indicating a data decompression error, or perhaps an IAMW37 I/O error message, or an IAMW17 error message indicating concurrent updates. Or perhaps you are receiving unexpected errors or results from the application, which can include receiving unexpected error codes from IAM while processing the dataset. These types of errors may indicate a problem with the dataset itself, however they more frequently arise due to software failures such as inadvertent storage corruption. Such errors frequently do not corrupt the file. For these situations, one would normally start by running an IAMRECVR DIAGNOSE operation to validate general dataset integrity, which then may be followed with a recovery or a reorganization of the dataset. If you suspect that IAM is not returning all of the records that are expected, it is critical to run an IAMRECVR RECOVER on the dataset before reorganizing the file or otherwise copying the data. Depending on the nature of the problem, IAMRECVR may be able to retrieve records that can not be retrieved by normal IAM processing that utilizes the index structure of the dataset.

10.87 CONTINUED . . .

IAMRECVR (Continued)

The second category of failure is where IAM is unable to open the dataset. Generally, an IAM error message will be displayed indicating the cause of the failure to open. If the problem is other than environmental, such as insufficient storage, then a dataset recovery will be necessary. Generally, this type of error is accompanied by an IAMW79 or an IAMW01 with an IAMW37 error message. These messages contain information about the point of failure during the open, so it is critical to save this message for diagnostic purposes. For this category of errors, it is best to start out with obtaining initial diagnostic information, and then proceed directly to use the IAMRECVR RECOVER command to copy the data records out of the file.

IAMRECVR will provide a report on any errors it may find, plus it will indicate how many records that it actually read. This record count can be used to compare with the count from other sources to determine if all the records have been retrieved. There may be some circumstances where an IAM dataset can not be opened for which IAMRECVR successfully performs the recovery, however finds no detectable errors. This is because IAMRECVR is not using the index structure of the dataset to read the data, so it may not be detecting the error condition.

DIAGNOSE

When errors are encountered when processing an IAM dataset, generally the first step is to validate the general dataset integrity with an IAMRECVR DIAGNOSE function. This is most useful for circumstances where you suspect that there might be a problem with an IAM dataset. For example, if a job has received an IAMW12 data decompress error, or an IAMW17 indicating concurrent updating, or an I/O error message processing the IAM dataset. Or, perhaps there has been a system failure and you just want to verify the integrity of the IAM dataset. This is done with the DIAGNOSE command.

DIAGNOSE will read the entire IAM dataset, validating basic file integrity. It will verify the following:

1. All data blocks can be physically read.
2. Records are in ascending key sequence within each data block.
3. All data blocks have valid structure.
4. All compressed records can be decompressed.
5. Records are in ascending key sequence within the prime area of the dataset.
6. Verify that there are no duplicate records within a data block.

DIAGNOSE will provide information on any errors it detects, along with a count of the number of records it was able to read from the dataset. The output will also include a report with basic information about the dataset that is quite similar to the IAMPRINT LISTCAT report. This information will help you identify if there is actually a problem within the dataset itself, and if so how much of the data will be recoverable.

10.87 CONTINUED . . .

EXA.MPLE A: IAMRECVR DIAGNOSE

The example below demonstrates how to run an IAMRECVR DIAGNOSE. The main requirements are providing a control card input on SYSIN that specifies the DIAGNOSE command, a DD statement with the name of DISKIN that allocates the IAM dataset to be diagnosed, and a SYSPRINT DD card for the printed output.

```
//DIAGNOSE EXEC PGM=IAMRECVR,REGION=0M
//SYSPRINT DD   SYSOUT=*
//DISKIN   DD   DSN=my.iam.dataset,DISP=SHR
//SYSIN    DD   *
      DIAGNOSE
/*
```

Figure 101: Example of an IAMRECVR DIAGNOSE (EX1087A)

The results of the DIAGNOSE process will provide information on the status of the dataset, and how many of the data records can be recovered. If problems were found, or difficulties are continuing with the dataset, then a recovery must be performed. Based on the information provided by the DIAGNOSE run, you can then determine what the best recovery method will be for this particular dataset. If it looks like IAMRECVR can retrieve all of the data records, or there are no other alternative recovery procedures available, then the IAMRECVR RECOVER command can be used to create a sequential output file containing the data.

EXAMPLE B: MULTIPLE DIAGNOSE

IAMRECVR can also DIAGNOSE multiple datasets in a single execution. This is achieved by specifying different DD names on separate DIAGNOSE commands using the FROMDDNAME keyword on the DIAGNOSE command, as illustrated by the following example.

```
//MULTDIAG EXEC PGM=IAMRECVR,REGION=0M
//SYSPRINT DD   SYSOUT=*
//FILE1    DD   DISP=SHR,DSN=my.iam.file1
//FILE2    DD   DISP=SHR,DSN=my.iam.file2
//FILE3    DD   DISP=SHR,DSN=my.iam.file3
//SYSIN    DD   *
  DIAGNOSE FROMDD=FILE1
  DIAGNOSE FROMDD=FILE2
  DIAGNOSE FROMDD=FILE3
/*
```

Figure 102: Example of Diagnosing multiple files (EX1087B)

OBTAINING DIAGNOSTIC INFORMATION

One of the important considerations in recovering an IAM dataset is that the corrupted dataset may be needed for problem determination and resolution. If it is at all possible, it is best to save the existing problem dataset where it is, and recover into a new dataset. If that is not possible, then the next best choice is to back up the dataset with FDR from Innovation Data Processing, or a comparable software product, such as DFSMSdss from IBM. Refer to the section on backing up IAM datasets for information and examples of how to obtain a backup copy with one of those products. Other types of backup copies may not preserve the exact image of the dataset, which can result in not being able to perform problem determination. An IDCAMS REPRO, or the output dataset from IAMRECVR RECOVER command will not be adequate for performing problem resolution.

10.87 CONTINUED . . .

EXAMPLE C: PRINT IDPINQ

Equally important is to obtain some diagnostic information. If you call for assistance due to a potentially damaged file, you will frequently will be asked to obtain various diagnostic information. The IAMRECVR PRINT command is used to help obtain some of this information. It will be quite helpful to run the job below prior to calling for technical assistance, as this is generally a diagnosis starting point. This job uses the PRINT command of IAMRECVR to print out the blocks within the dataset that describe the characteristics and physical structure of the file. A LISTCAT ALL is also being performed, as that will print out additional information on the volume(s) that the dataset resides.

```
//PRINTIDP EXEC PGM=IAMRECVR,REGION=4096K
//SYSPRINT DD   SYSOUT=*
//DISKIN   DD   DSN=my.iam.dataset,DISP=SHR
//SYSIN    DD   *
  PRINT    IDPINQ
/*
//LISTCAT  EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=*
//IAMPRINT DD   SYSOUT=*
//SYSIN    DD   *
  LISTCAT ENT(my.iam.dataset) ALL
/*
```

Figure 103: PRINT IDPINQ Example (EX1087C)

EXAMPLE D: PRINTING BLOCKS

With information from the above PRINT or DIAGNOSE, you may also be asked to print out selected blocks from the IAM dataset. The IAM Technical Support representative will inform you if this is needed, and provide you with the block range(s) to be printed. This is accomplished with a different flavor of the PRINT command. In this case, a starting block number is specified by the FBLK= (from block) keyword, and the number of blocks to print is specified with the MAXBLKS= keyword. You can specify multiple PRINT commands in the same execution of IAMRECVR, each of which will specify different ranges of blocks to be printed. The example below shows how this is done.

```
//PRINTBLK EXEC PGM=IAMRECVR,REGION=4096K
//SYSPRINT DD   SYSOUT=*
//DISKIN   DD   DSN=my.iam.dataset,DISP=SHR
//SYSIN    DD   *
  PRINT FBLK=65,MAXBLKS=4
  PRINT FBLK=100,MAXBLKS=1
/*
```

Figure 104: Example of printing out blocks of an IAM Dataset (EX1087D)

10.87 CONTINUED . . .

RECOVER To perform the recovery, the IAMRECVR RECOVER command is used to obtain a sequential copy of the data. If there are records in the Extended Overflow (or Independent Overflow for Compatible format files), the sequential output file must be sorted. The RECOVER command will invoke the sort product you have installed at your installation. The JCL you provide must specify whatever DD cards are needed for the sort. Frequently this requires specification of sort work space, with three or more SORTWK0x DD cards. This sort work space must be adequate enough to handle the amount of data that is contained within the dataset. The recover step is followed by a step to define a new cluster and reload the data with IDCAMS REPRO. The reload step also includes renaming the datasets.

SPANNED RECORDS To perform a recovery on files with spanned records, there are additional considerations. On the step executing the IAMRECVR RECOVER command, an additional DD statement is required. That DD statement is SPANOUT. This DD statement defines a file on tape or disk that will contain those records that are actually spanned, that is too large to fit within a single block. After performing the reload of the dataset from the TAPEOUT file, an additional step is required for the spanned records. That step is to run the IAMRECVR APPLY command with the SPANNED keyword. That will update the recover file with the spanned records from the SPANOUT file. Further information on using these keywords is in section 45.

10.87 CONTINUED . . .

EXAMPLE E: BASIC RECOVER

The example below performs both the RECOVER step and the REPRO step. First, the RECOVER command of IAMRECVR is executed to create a sequential file containing all of the data records from the IAM dataset. If that process is successful, then IDCAMS is executed. With IDCAMS, a new dataset is defined, using the original dataset as a model, and then loaded with the recovered data. This is followed by renames of the datasets. Note the use of the IDCAMS IF and CANCEL commands, which is done to preserve the original dataset in case a failure occurs during the IDCAMS processing.

For the execution of the IAMRECVR program, the DISKIN DD specifies the IAM dataset, and the TAPEOUT DD specifies the new sequential dataset. The SYSPRINT and SYSIN DD cards are required by IAMRECVR. The SORTWK0x and SYSOUT DD statements are provided for the SORT.

For the execution of the IDCAMS step, the SYSPRINT and SYSIN DD are required. The sequential file created by the RECOVER process in the prior step is included with a DD name of INFILE. The IAMINFO DD is optional, but recommended to obtain the run time report for the file load. Note that there is a DD statement, OLDIAMDS, for the original IAM dataset that is otherwise not referenced. This is done to hold the ENQ on the original dataset name until the recovery process is complete. Proper caution should be used in constructing the job stream to insure that the data is preserved.

WARNING: If you are running with a version of IAMRECVR prior to Version 6.4 and you need to Delete and Define the IAM dataset, use the JCL for Example F. If there are duplicate keys in the dataset, the records will be lost when the dataset is deleted. In Version 8.0, IAMRECVR will set a return code of 8 when duplicate keys are found, and you are not logging or ignoring the duplicate keys.

```
//RECOVER   EXEC PGM=IAMRECVR,REGION=4096K
//SYSPRINT DD   SYSOUT=*
//SYSOUT   DD   SYSOUT=*
//DISKIN   DD   DISP=OLD,DSNAME=my.iam.dataset
//TAPEOUT  DD   DSN=my.seq.dataset,DISP=(,CATLG),
//     UNIT=SYSDA,SPACE=(CYL,(20,10))
//SORTWK01 DD   UNIT=SYSDA,SPACE=(CYL,(20,10))
//SORTWK02 DD   UNIT=SYSDA,SPACE=(CYL,(20,10))
//SORTWK03 DD   UNIT=SYSDA,SPACE=(CYL,(20,10))
//SYSIN    DD   *
  RECOVER
/*
//LOADNEW   EXEC PGM=IDCAMS,COND=(0,NE)
//SYSPRINT DD   SYSOUT=*
/IAMINFO   DD   SYSOUT=*
//INFILE   DD   DSN=my.seq.dataset,DISP=OLD
//OLDIAMDS DD   DSN=my.iam.dataset,DISP=OLD
//SYSIN    DD   *
  DELETE my.newiam.dataset
  DELETE my.newiam.dataset NOSCRATCH
  SET MAXCC=0
  DEFINE CLUSTER(NAME(my.newiam.dataset)   -
      OWNER($IAM)                          -
      MODEL(my.iam.dataset))
  IF MAXCC NE 0 THEN CANCEL
  REPRO INFILE(INFILE) ODS(my.newiam.dataset)
  IF MAXCC NE 0 THEN CANCEL
  ALTER my.iam.dataset NEWNAME(my.badiam.dataset)
  IF MAXCC NE 0 THEN CANCEL
  ALTER my.newiam.dataset NEWNAME(my.iam.dataset)
  LISTCAT ENT(my.iam.dataset) ALL
  LISTCAT ENT(my.badiam.dataset) ALL
/*
```

Figure 105: Basic Dataset Recovery Example (EX1087E)

10.87 CONTINUED . . .

DUPLICATE KEYS

One of the circumstances that may occur is that records with duplicate keys are discovered by the RECOVER process after the sort has been done, while the output sequential dataset is being written. This circumstance does not necessarily represent a data integrity problem with the file. When a record is updated, the length of the record may be changed either by the application program itself, or by IAM if the updated record compresses differently. If there was an increase in the record length as a result of the update, the record may no longer fit within the current block that it resides, so it is moved by IAM to Overflow. Without Variable Overflow because the maximum length is reserved for a record once it is moved to an overflow block it will stay in that block. With Variable Overflow, the record may need to be moved to a different overflow block. IAM will first write out the updated record within the block it was moved to, and then subsequently write out the original block with the old record deleted. If a failure occurs that prevents the proper closing of the dataset, the second write might not yet have been done resulting with the record existing in both blocks. Failures that may result in this condition include MVS failures resulting in an IPL without proper application shutdown, using the MVS FORCE command to cancel an updating job from the system, or other types of address space failures, or power outage. Files that were opened for update during such a failure should be reorganized or recovered as soon as possible after such a failure. Unfortunately, such failures also prevent the file statistics from being updated as well, so accurate information may not be reflected in the statistics particularly for the actual record count.

Other possibilities for duplicate keys include sharing the IAM dataset for concurrent updating or software failures that caused storage corruption. For these type of duplicates, you may need to examine which of the duplicate records you want to have in the recovered dataset. This can be accomplished by editing the LOG dataset that is created by IAMRECVR, and then running the APPLY step.

During normal IAM processing, the first duplicate record condition is not a problem as long as the record is not deleted. This is because with Enhanced format files, the record will always be moved to a higher relative block than it existed in before the update. So, the valid record will always be the record in the highest block. (Note that for Compatible format files, the situation is reversed because the Overflow area is at the physical beginning of the dataset.) For a recovery using IAMRECVR, a different procedure than the basic one shown in the preceding example must be used.

10.87 CONTINUED . . .

EXAMPLE F: RECOVER WITH DUPLICATE KEYS

The first change to the original example is that the SORT must be told to pass records with equal keys back in the same order that they were passed to the SORT. This is done with the EQUALS option for DFSORT and SYNCSORT. For DFSORT, this option is specified by a control card input using the DFSPARM DD. For SYNCSORT, a $ORTPARM DD card is used. In the example, both are included. The next change for Enhanced format files is to specify a LOG dataset and indicating on the RECOVER command that records with duplicate keys are to be logged, (i.e. specify DUP=LOG on the RECOVER command). The first record of any specific key value will always be written to the normal sequential output dataset. Any subsequent records with the same key will be written out to the LOG dataset. Then, the normal reload is done for the dataset.

Then, for Enhanced format files only, the records in the LOG dataset are copied into the IAM dataset with an IAMRECVR APPLY statement. Note that you could also use an IDCAMS with the REPRO REPLACE statement. See Example G for using the REPRO REPLACE instead of the IAMRECVR APPLY. The advantage of using the IAMRECVR APPLY is that it will print out the keys of the records that are being replaced by the apply operation.

```
//RECOVER   EXEC PGM=IAMRECVR,REGION=4096K
//SYSPRINT DD   SYSOUT=*
//SYSOUT   DD   SYSOUT=*
//DISKIN   DD   DISP=OLD,DSNAME=my.iam.dataset
//TAPEOUT  DD   DSN=my.seq.dataset,DISP=(,CATLG),
//     UNIT=SYSDA,SPACE=(CYL,(20,10))
//LOG      DD   DSN=my.duprec.dataset,DISP=(,CATLG),
//     UNIT=SYSDA,SPACE=(CYL,(2,1))
//SORTWK01 DD   UNIT=SYSDA,SPACE=(CYL,(20,10))
//SORTWK02 DD   UNIT=SYSDA,SPACE=(CYL,(20,10))
//SORTWK03 DD   UNIT=SYSDA,SPACE=(CYL,(20,10))
//$ORTPARM DD   *
EQUALS
/*
//DFSPARM  DD   *
EQUALS
/*
//SYSIN    DD   *
  RECOVER  DUP=LOG
/*
//LOADNEW  EXEC PGM=IDCAMS,COND=(0,NE)
//SYSPRINT DD   SYSOUT=*
//IAMINFO  DD   SYSOUT=*
//INFILE   DD   DSN=my.seq.dataset,DISP=OLD
//SYSIN    DD   *
  DELETE my.iam.dataset
   IF MAXCC NE 0 THEN CANCEL
  DEFINE CLUSTER(NAME(my.iam.dataset)        -
      OWNER($IAM)                            -
      VOL(myvol) CYL(20 10)                  -
      RECORDSIZE(300 1000)                   -
      KEYS(16 0)                             -
      FREESPACE(10 10)                       -
      SHAREOPTIONS(2 3)        )
 IF MAXCC NE 0 THEN CANCEL
 REPRO INFILE(INFILE) ODS(my.iam.dataset)
 LISTCAT ENT(my.iam.dataset) ALL
/*
//APPLY    EXEC PGM=IAMRECVR,REGION=0M
//SYSPRINT DD   SYSOUT=*
//IAMINFO  DD   SYSOUT=*
//LOG      DD   DSN=my.duprec.dataset,DISP=OLD
//VSAMOUT  DD   DSN=my.iam.dataset,DISP=OLD
//SYSIN    DD   *
 APPLY OUT=VSAM
/*
```

Figure 106: Example of Recovering Dataset With Duplicate Keys (EX1087F)

10.87 CONTINUED . . .

EXAMPLE G: RECOVER WITH REUSE

In this next example, rather than deleting and redefining the dataset, it is copied into with a REPRO REUSE. If that is successful, then the duplicates, if any, are copied into the IAM dataset with another REPRO, but this time with REPLACE.

```
//RECOVER   EXEC PGM=IAMRECVR,REGION=4096K
//SYSPRINT DD   SYSOUT=*
//SYSOUT   DD   SYSOUT=*
//DISKIN   DD   DISP=OLD,DSNAME=my.iam.dataset
//TAPEOUT  DD   DSN=my.seq.dataset,DISP=(,CATLG),
//     UNIT=SYSDA,SPACE=(CYL,(20,10))
//LOG      DD   DSN=my.duprec.dataset,DISP=(,CATLG),
//     UNIT=SYSDA,SPACE=(CYL,(2,1))
//SORTWK01 DD   UNIT=SYSDA,SPACE=(CYL,(20,10))
//SORTWK02 DD   UNIT=SYSDA,SPACE=(CYL,(20,10))
//SORTWK03 DD   UNIT=SYSDA,SPACE=(CYL,(20,10))
//$ORTPARM DD   *
EQUALS
/*
//DFSPARM  DD   *
EQUALS
/*
//SYSIN    DD   *
  RECOVER  DUP=LOG
/*
//LOADNEW  EXEC PGM=IDCAMS,COND=(0,NE)
//SYSPRINT DD   SYSOUT=*
//IAMINFO  DD   SYSOUT=*
//IAMFILE  DD   DSN=my.iam.dataset,DISP=OLD
//INFILE   DD   DSN=my.seq.dataset,DISP=OLD
//DUPFILE  DD   DSN=my.duprec.dataset,DISP=OLD
//SYSIN    DD   *
  REPRO INFILE(INFILE) OUTFILE(IAMFILE) REUSE
  IF MAXCC NE 0 CANCEL
  REPRO INFILE(DUPFILE) OUTFILE(IAMFILE) REPLACE
 LISTCAT ENT(my.iam.dataset) ALL
/*
```

Figure 107: Example of recover with REPRO (EX1087G)

10.88 IAM JOURNAL AND RECOVERY

This section provides information on using IAM journaling and recovery for IAM datasets not processed by IAM RLS. For the journaling and recovery offered for datasets processed with IAM RLS, refer to Section 20 on Using IAM RLS.

OVERVIEW

For Enhanced Format IAM datasets without spanned records, IAM offers an optional journaling facility along with an associated recovery capability. IAM journaling is activated through IAM Overrides for the datasets that you select. The IAM journaling facility provides a tool to capture in a separate dataset the images of records prior to being updated, called before images, and / or images of records after being updated, called after images. Once a journal has been created, it can be used by the provided utility program IAMJREST to recover the IAM dataset to a particular point in time. There are two types of recovery possible, called forward recovery and backward (or backout) recovery. Forward recovery is done by restoring a backup copy of the affected IAM dataset, then applying the after images from the journal dataset(s) to the IAM dataset. In other words, the recovery proceeds forward from a particular point in time to an identified point in time, prior to the failure. A backout recovery does the opposite function. Starting with a dataset that has been updated by various jobs, it provides a capability to backout the updates from selected jobs. This is accomplished by updating the dataset from the log using the before images.

WHY USE IAM JOURNALING

The main reason to use the IAM journaling feature is to improve data availability for very large datasets. This is accomplished because the enhanced recovery capabilities can reduce the frequency of backing up the entire dataset. Rather than backing up large IAM datasets every day, they can be backed up less frequently, perhaps just once a week. Only the journal containing the updated data is backed up on a daily basis. By backing up less data, the amount of time a dataset is not available for update processing is reduced. The amount of time saved by not doing full backups each night can be quite substantial.

In case of a media failure, the dataset is recovered by restoring the dataset from the last full backup, then executing a journal forward recovery using IAMJREST. For failures when a batch job abends, or otherwise fails, the updates made by the failing batch job can be backed out by IAMJREST, then the batch job processing can be restarted. If need be, the updates from multiple jobs steps and jobs can be backed out. The backout facility may be able to save a lot of time when recovering from typical job abends, by eliminating the need to restore datasets and rerun an entire sequence of batch jobs. Additional savings are possible by eliminating the need for batch jobs to perform separate backup of critical files prior to starting an update process.

PREPARING FOR JOURNALING

Before turning on IAM journaling for a dataset, you need to take a few steps in preparation. First, you must decide what types of recoveries you are going to want to perform, as this choice impacts the type of records you will need to have written to the journal dataset. If you are planning on both the forward and backward recoveries, you will need to be collecting both before and after images. If you are planning on just forward recoveries, then only the after image records are collected. Conversely, if you are only planning on only backout recoveries, then you only need to collect before images. You can also choose between journaling all update activity against a file, or only journaling for some selected jobs. If you are journaling for all or most of the update activity on the IAM dataset, then you should activate journaling by specifying the IAM CREATE JRNAD= override when the file is defined or loaded. The journaling option you select will be retained with the dataset, and used on each access to the dataset. If you are only planning on using the journaling for a few selected jobs to be able to backout their updates, then the journaling can be specified on the selected jobs steps using the IAM ACCESS JRNAD= override. As operands on the JRNAD keyword, you specify the types of records that you need to have written to the journal. Valid values are:

- BEFORE for before images,
- AFTER for only updated record images, or
- BOTH to collect BEFORE and AFTER images.

Using BOTH will provide the most flexibility, but also incurs the largest overhead in terms of space required and journal I/O activity. When estimating the space, keep in mind that IAM does NOT journal during file load or reorganization, so there will not be a "second copy" in the journal of the entire dataset.

10.88 CONTINUED . . .

MANAGING THE JOURNAL DATASET

You are responsible for allocating and managing the journal dataset. The next step is to decide how you want to manage your IAM journal datasets. The IAM Journal datasets must reside on DASD while they are being actively used for journaling. They can reside on either tape or disk when performing the recovery phase. If you are opting for the full journaling, then you will need permanent DASD datasets for the journals. Typically, the journal datasets will be backed up daily using IEBGENER. The reason for using IEBGENER is so that the back up copy of the LOG dataset can be used directly as input to the IAMJREST utility. That would not be possible if the dataset was backed up by DASD management software products, such as FDR/ABR. The backup copy can take the form of an accumulation dataset, or as a separate dataset each day, perhaps by using a GDG.

To help minimize both the amount of DASD space for the journal and the journal backup time, you will probably want to empty the journal dataset(s) after they are backed up. This can be easily accomplished by running program IEBGENER with an empty input dataset. Or, you might decide that it is more critical to minimize the time it takes to set up the recovery. In that case you would want to accumulate the journal data within the journal dataset itself, containing all the data since the last full dataset backup.

If you are just going to do journaling for batch update jobs to provide a backout recovery capability, you have some different choices. You can either go with a permanently allocated journal dataset, or allocate one at the beginning of the batch job stream. In either case, you will most likely want to backup the journal dataset at the end of the batch stream. If you are using a permanent journal dataset, you will want to empty it prior to or at the beginning of the job stream that is being journalled.

ESTIMATING JOURNAL DASD SPACE

With the above decisions made, you can next estimate the amount of DASD space that will be needed for your journal dataset. To make the estimate, you will need some statistical information that can be found in IAM LISTCAT or IAMINFO reports. These reports should cover the typical length of time that the data will be residing in the journal, whether that is just for a day, a week, or through a batch job stream. The important numbers are the number of inserts, updates and deletes for the length of time in question, as this will be used to calculate the number of records that are written to the journal. To estimate the number of records, which we will call R, select one of the following calculations based on the types of records that are being collected:

- For both BEFORE and AFTER Images: **R** = (2 * (updates + deletes)) + inserts.
- For only BEFORE or AFTER Images: **R** = updates+deletes+inserts.

The IAM journal will typically use _ track blocking, which will be 27998 when residing on a 3390, or 23476 when residing on a 3380, and using variable length records and blocks. The maximum record length is the defined user record size plus 52 bytes for the header information on each journal record. If the maximum record size exceeds the _ track block size, then a block size of 32760 will be used. To play it absolutely safe, you should estimate your journal space requirements using the maximum record length. However, for files that have a very large maximum record length, for which you know that the actual record length is considerably smaller, you can use the smaller record length. Select the record size that seems most appropriate, and add 44 bytes to that length for the header information. Divide the expected block size-4 by the record size to get the average number of journal records per block, dropping any fraction. Multiply the result by 30 if using _ track blocking, or by 15 if the larger block size of 32760 must be used. That will yield the number of journal records per cylinder, which we will call C.

The estimated number of cylinders will be R / C, which is to be rounded up. Or in words, the estimated number of log records divided by the number of records per cylinder yields the number of cylinders. Use this value for the primary space allocation of the journal dataset. Then use some fraction in the range of 10% to 25% as the secondary space value for your journal dataset allocation.

10.88 CONTINUED . . .

EXAMPLE OF JOURNAL SPACE CALCULATION

As an example, let's say that we have a file that typically has 15,000 updates, 1,000 inserts, and 500 deletes in a day. Our plan is to collect both before and after images, enabling both forward and backout recovery. The journal will be backed up and then emptied each day after the online system comes down, prior to starting the batch update runs. The estimated number of daily journal records will be:

R = (2 * (15,000 + 500)) + 1,000 = 32,000.

The maximum defined user record length is 256, so the maximum journal record length is (256 + 52) = 308. The journal is going to reside on a 3390, which has a _ block size of 27,998. Dividing (27,998-4) by 308 = 90 log records per block. Next, multiply the number of log records per block times the number of blocks per cylinder (which is 30), to come up with 2,790 log records per cylinder.

C = ((27998-4) / (256 + 52)) * 30 = 2,700.

The number of cylinders required is **R / C** or 32,000 / 2,700 = 12.

JOURNAL DATASET NAME

The name of the journal dataset is the dataset name (or cluster name) of the IAM dataset, with the characters '.LOG' appended to the end. If the length of the dataset name exceeds 40 bytes, then the end will be overlaid with the '.LOG' character string and be 44 bytes long, except when the 40th byte is a '.'. In that case, the dataset name will only be 43 bytes long, still ending with the '.LOG' literal.

JOURNAL ALLOCATION CONSIDERATIONS

There are a few considerations for doing the actual journal allocation. First, to maximize the recovery potential, and to avoid I/O contention between the journal and the actual dataset, the journal dataset must be on a different volume. It would be best for the journal dataset to be on a device that is on a different channel and controller if that is possible. Secondly, make sure to pick a volume that has sufficient DASD space available to allow the journal dataset to go into extents if necessary. Thirdly, do not specify any DCB characteristics, such as LRECL, BLKSIZE, or RECFM. The IAM journal program will set those values automatically the first time the journal dataset is used.

EXAMPLE A: DEFINING AN IAM FILE WITH JOURNALING AND A LOG DATASET

The example JCL below demonstrates defining an IAM dataset with the journaling option requested on the IAM CREATE Override statement. Note that JRNAD=BOTH is specified, indicating journaling of both before and after images. This option provides for both forward and backout recoveries. The journal dataset is also being allocated in the same job step, using the SPACE value calculated in the above example calculation.

```
//IAMDEFIN  EXEC PGM=IDCAMS
//SYSPRINT  DD   SYSOUT=*
//JOURNAL   DD   DSN=MY.IAM.KSD.LOG,UNIT=SYSDA,DISP=(,CATLG),
//     SPACE=(CYL,(12,6)),VOL=SER=MYVOL2
//IAMOVRID  DD   *
      CREATE DD=&ALLDD,JRNAD=BOTH
/*
//SYSIN     DD   *
  DEFINE CLUSTER            -
    (NAME(MY.IAM.KSD)       -
    OWNER($IAM)             -
    VOLUMES(MYVOL1)         -
    CYL(60 6)               -
    RECORDSIZE(100 256)     -
    KEYS(24 8)              -
    FREESPACE(5 20)         -
    SHAREOPTIONS(2 3)       -
    REUSE  )
   LISTCAT ENT(MY.IAM.KSD) ALL
/*
```

Figure 108: Example of Defining an IAM Dataset with Journaling (EX1088A)

10.88 CONTINUED . . .

EXAMPLE B: SETTING UP GDG FOR JOURNAL BACKUPS

To continue with the above example, the next step is setting up whatever JCL is needed for management of the log data. For the IAM dataset being journalled, the backup frequency had been once a day. With the IAM journal being active for this dataset, the backup frequency will be changed to once a week. The journal will be backed up daily, to a GDS, and then emptied. As part of the weekly dataset backup, we will accumulate the daily journals for the past week, into a single dataset on tape, which will be the weekly journal GDS. Step 2 is then to set up the GDG for the journal backups, which can be done as shown below:

```
//IAMDEFIN  EXEC PGM=IDCAMS
//SYSPRINT  DD   SYSOUT=*
//SYSIN     DD   *
  DEFINE GENERATIONDATAGROUP                  -
    (NAME(MY.IAM.KSD.DAILY.JOURNAL)           -
    LIMIT(7) SCRATCH )
  DEFINE GENERATIONDATAGROUP                  -
     NAME(MY.IAM.KSD.WEEKLY.JOURNAL)          -
     LIMIT(54))
/*
```

Figure 109: Defining GDG for Journal Backups (EX1088B)

EXAMPLE C: DAILY JOURNAL BACKUP JOB

The daily journal backup job will run six days a week. This job will not only make a copy of the journal dataset to the new daily journal backup GDG, but will also empty the journal file so that the most it will contain is one day's worth of data. The following example will show how this can be accomplished:

```
//BKUPJRNL EXEC PGM=IEBGENER
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   DUMMY
//SYSUT2   DD   DSN=MY.IAM.KSD.DAILY.JOURNAL(+1),
//     DISP=(,CATLG),SPACE=(CYL,(12,6)),UNIT=SYSDA,
//     DCB=(MY.IAM.KSD.LOG),VOL=SER=MYVOL3
//SYSUT1   DD   DSN=MY.IAM.KSD.LOG,DISP=OLD
//EMPTYJRN EXEC PGM=IEBGENER,COND=(0,NE,BKUPJRNL)
//SYSPRINT DD   SYSOUT=*
//SYSIN    DD   DUMMY
//SYSUT1   DD   DUMMY,DCB=(MY.IAM.KSD.LOG)
//SYSUT2   DD   DSN=MY.IAM.KSD.LOG,DISP=OLD
```

Figure 110: Example of Daily Journal Backup (EX1088C)

10.88 CONTINUED . . .

EXAMPLE D: WEEKLY JOURNAL BACKUP

The weekly journal backup job should be run as part of the same job that performs the backup of the base IAM dataset. Preferably as the step immediately after the dataset backup. If time is critical, the weekly backup of the journal datasets can be done concurrent with the dataset backup. This job will copy the previous six daily backups along with any current data in the journal, and subsequently empty the journal dataset itself.

```
//WBKUPJRN   EXEC PGM=IEBGENER
//SYSPRINT   DD   SYSOUT=*
//SYSIN      DD   DUMMY
//SYSUT2     DD   DSN=MY.IAM.KSD.WEEKLY.JOURNAL(+1),
//      DISP=(,CATLG),UNIT=CART
//SYSUT1     DD   DSN=MY.IAM.KSD.DAILY.JOURNAL(-5),DISP=OLD
//           DD   DSN=MY.IAM.KSD.DAILY.JOURNAL(-4),DISP=OLD
//           DD   DSN=MY.IAM.KSD.DAILY.JOURNAL(-3),DISP=OLD
//           DD   DSN=MY.IAM.KSD.DAILY.JOURNAL(-2),DISP=OLD
//           DD   DSN=MY.IAM.KSD.DAILY.JOURNAL(-1),DISP=OLD
//           DD   DSN=MY.IAM.KSD.DAILY.JOURNAL(0),DISP=OLD
//           DD   DSN=MY.IAM.KSD.LOG,DISP=OLD
//EMPTYJRN   EXEC PGM=IEBGENER,COND=(0,NE,WBKUPJRN)
//SYSPRINT   DD   SYSOUT=*
//SYSIN      DD   DUMMY
//SYSUT1     DD   DUMMY,DCB=(MY.IAM.KSD.LOG)
//SYSUT2     DD   DSN=MY.IAM.KSD.LOG,DISP=OLD
```

Figure 111: Weekly Backup to Tape of Journal Data (EX1088D)

JOURNAL RECOVERY PROCEDURES

Now that the journaling is all in place, along with the proper backup procedures, the recovery process from the IAM journal can be examined. Recovery with the journal is provided for by the IAMJREST program. The IAMJREST program uses as input the journal dataset(s), the base IAM dataset, and some control parameters directing its operation. The output of IAMJREST is the updated IAM dataset. This section will provide the basics of using IAMJREST, along with examples. For additional information in IAMJREST, refer to section 47 of the manual, which has a complete description of the IAMJREST program.

FORWARD RECOVERY

There are two different types of recovery procedures that can be performed by IAMJREST. The first type is a forward recovery. A forward recovery will typically be used to recover a dataset if the device it resides on is damaged or not operational, or if the dataset has been seriously corrupted. A forward recovery is one where first the base dataset is restored, then the IAMJREST utility is executed to apply the updates, inserts, and deletes from the journal to the base dataset. For a forward recovery to occur, the journal must contain AFTER images for all the jobs that updated the dataset. This is accomplished by specifying JRNAD=BOTH or JRNAD=AFTER on the IAM CREATE override statement.

10.88 CONTINUED . . .

EXAMPLE E: FORWARD RECOVERY WITH RESTORE

Shown below is an example of performing a forward recovery. In the example, it is presumed that a failure occurred on the third day after the last backup. So, there are the two daily backups of the journal file, plus the current journal file that will be used for the recovery. These three datasets will be input to the recovery procedure. The dataset being recovered will first be deleted and defined on a new volume, then it will be reloaded from the backup tape. After the restore, IAMJREST is executed with the three input files to perform the forward recovery. The failure occurred will running the job BATCHUP2. The recovery will be performed for all update jobs that ran up to and including job BATCHUP1. Once any other datasets are recovered, the BATCHUP2 job can be restarted.

```
//DELDEFIN  EXEC PGM=IDCAMS
//SYSPRINT  DD   SYSOUT=*
//IAMOVRID  DD   *
      CREATE DD=&ALLDD,JRNAD=BOTH
/*
//SYSIN     DD   *
  DELETE MY.IAM.KSD CLUSTER
  IF MAXCC NE 0 THEN -
      DELETE MY.IAM.KSD NOSCRATCH
      SET MAXCC = 0
  END
  DEFINE CLUSTER          -
   (NAME(MY.IAM.KSD)      -
   OWNER($IAM)            -
   VOLUMES(NEWVOL)        -
   CYL(60 6)              -
   RECORDSIZE(100 256)    -
   KEYS(24 8)             -
   FREESPACE(5 20)        -
   SHAREOPTIONS(2 3)      -
   REUSE  )
  LISTCAT ENT(MY.IAM.KSD) ALL
/*
//RESTORE  EXEC  PGM=IDCAMS,COND=(0,NE)
//SYSPRINT DD    SYSOUT=*
//BACKUP   DD    DSN=MY.IAM.KSD.BACKUP,DISP=OLD
//IAMFILE  DD    DSN=MY.IAM.KSD,DISP=OLD
//IAMINFO  DD    SYSOUT=*
//SYSIN    DD    *
      REPRO INFILE(BACKUP) OUTFILE(IAMFILE) REUSE
      LISTCAT ENT(MY.IAM.KSD) ALL
/*
//RECOVER  EXEC  PGM=IAMJREST,REGION=64M,COND=(0,NE)
//SYSPRINT DD    SYSOUT=*
//SYSOUT   DD    SYSOUT=*
//SORTWK01 DD    UNIT=SYSDA,SPACE=(CYL,(50,10))
//SORTWK02 DD    UNIT=SYSDA,SPACE=(CYL,(50,10))
//SORTWK03 DD    UNIT=SYSDA,SPACE=(CYL,(50,10))
//IAMFILE  DD    DSN=MY.IAM.KSD,DISP=OLD
//IAMJRNL  DD    DSN=MY.IAM.KSD.DAILY.JOURNAL(-1),DISP=OLD
//         DD    DSN=MY.IAM.KSD.DAILY.JOURNAL(0),DISP=OLD
//         DD    DSN=MY.IAM.KSD.LOG,DISP=OLD
//IAMINFO  DD    SYSOUT=*
//SYSIN    DD    *
      RESTORE FORWARD,TOJOB=BATCHUP1,TODATE=1998182
/*
```

Figure 112: Forward Recovery Preceded by a Restore (EX1088E)

10.88 CONTINUED . . .

BACKOUT RECOVERY The other type of recovery that can be performed is a backout (or backwards) recovery. This type of recovery backs out updates to a dataset from the specified jobs or job steps. To perform a backout recovery, the journal must include BEFORE images. BEFORE images are written to the journal when either JRNAD=BOTH or JRNAD=BEFORE have been specified on the IAM overrides. A forward recovery can be done to restore a dataset to the same point that can be done with a backout recovery. The backout recovery however will generally be much faster than a forward recovery. This is because the dataset does not have to be restored, and eliminates having to apply updates from potentially several days. Backout recoveries are also ideal if you only need to use IAM journaling to provide recovery for a few jobs or a job stream.

EXAMPLE F: BACKOUT RECOVERY Shown below is an example of a backout recovery. In this recovery, the all updates for job BATCHUPD that was run on June 30, 1998 (1998.181) are removed from the IAM dataset.

```
//RECOVER   EXEC   PGM=IAMJREST,REGION=64M,COND=(0,NE)
//SYSPRINT DD     SYSOUT=*
//SYSOUT   DD     SYSOUT=*
//SORTWK01 DD     UNIT=SYSDA,SPACE=(CYL,(50,10))
//SORTWK02 DD     UNIT=SYSDA,SPACE=(CYL,(50,10))
//SORTWK03 DD     UNIT=SYSDA,SPACE=(CYL,(50,10))
//IAMFILE  DD     DSN=MY.IAM.KSD,DISP=OLD
//IAMJRNL  DD     DSN=MY.IAM.KSD.DAILY.JOURNAL(-1),DISP=OLD
//         DD     DSN=MY.IAM.KSD.DAILY.JOURNAL(0),DISP=OLD
//         DD     DSN=MY.IAM.KSD.LOG,DISP=OLD
//IAMINFO  DD     SYSOUT=*
//SYSIN    DD     *
      RESTORE BACKOUT,JOBNAME=BATCHUPD,FROMDATE=1998181
/*
```

Figure 113: An Example of a Backout Recovery (EX1088F)

EXAMPLE G: IDENTIFYING THE CONTENTS OF A JOURNAL If you are unsure of what jobs or job steps have been journalled, the IAMJUTIL provides the SCAN capability. This command will read the specified journal dataset, and print a line identifying each of the job steps that have records on the file. Included with that information is the time and date. This information may be helpful when setting up to run a recovery. The SCAN command also has a DETAIL operand, which provides a way to determine the complete contents of the journal dataset, should that be needed. Normally though, it is expected that the summary information identifying all the job steps that have updated the dataset will be sufficient information.

```
//RECOVER   EXEC  PGM=IAMJUTIL
//SYSPRINT  DD    SYSOUT=*
//IAMJRNL   DD    DSN=MY.IAM.KSD.LOG,DISP=OLD
//SYSIN     DD    *
      SCAN  SUMMARY
/*
```

Figure 114: Example of Identifying Contents of Journal (EX1088G)

THIS PAGE WAS INTENTIONALLY LEFT BLANK

20.01 IAM RECORD LEVEL SHARING

OVERVIEW IAM Version 8.0 introduces IAM Record Level Sharing for a single system. With IAM RLS, concurrently running batch jobs, TSO users, and CICS regions running on the same LPAR can update IAM files, if they are accessed through IAM RLS. IAM is different from VSAM/RLS in that rather than having an application such as CICS indicate it is using IAM RLS, IAM will automatically select those datasets that should be processed by IAM RLS. Eligibility for processing under IAM RLS is determined during OPEN processing, on a file-by-file basis. When an eligible IAM dataset is opened, all of the I/O requests for that dataset are passed to the IAM RLS address space for processing, and the results returned to the requesting address space. IAM RLS provides the following capabilities:

- Ability to share IAM files for update between concurrently executing batch jobs, TSO users, and CICS regions.
- Locking during update processing is performed at the record level, utilizing the IAM RLS record locking facility.
- For recoverable files, record locks are held until the job or transaction terminates, or the job or transaction issues a SYNCPOINT.
- Potential deadlocks within the resources managed by IAM RLS are detected, and the associated request will be failed to prevent a deadly embrace.
- Journaling and recovery capabilities are provided for datasets processed under IAM RLS. Journaling can be done either to sequential datasets, or to the z/OS System Logger.
- Ability to dynamically back out updates made to files processed by IAM RLS when the batch job step that performed those updates abends.
- A callable batch syncpoint process is provided, to prevent batch programs from locking out access to large portions of recoverable files when mass updates are being performed.
- An ISPF interface to monitor activity within the IAM RLS address space, and issue some of the available operator commands.
- Depending on the RLS selection criteria implemented at an installation, there generally will be little or no JCL changes required to use IAM RLS. For CICS, the installation and activation of certain exits is required, as documented in the IAM RLS CICS Considerations in Section, 20.50. For batch jobs that perform a large volume of updates to recoverable files being processed by IAM RLS, the implementation of the IAM Batch Syncpoint, and appropriate restart capabilities are highly recommended.

20.01 CONTINUED . . .

RECORD LOCKING For the single system record level sharing, IAM utilizes its own record level lock manager. Records are locked by their key within each IAM dataset being processed by IAM RLS. The IAM lock manager does check for potential deadlocks within the scope of the IAM datasets that it is managing. If waiting on the record lock for the current request will result in a deadlock, the request will be failed with a logical error. The trigger for IAM to release a record lock will depend on the environment in which the lock was requested.

For CICS transactions, records are locked by transaction identification. IAM assumes that the IAM files accessed under CICS are recoverable, and therefore holds the record lock(s) until either a SYNCPOINT is executed, or the transaction ends. If the IAM file is defined with the explicit option of JRNAD=NONE or the IDCAMS parameter LOG(NONE), then IAM will assume that the file is not recoverable under CICS. When an IAM file is detected as being not recoverable under CICS, then the record lock is only held from the time of the GET for UPDATE until the record is actually updated or erased, or for records being added, only for the duration of the actual add processing.

For other than CICS processing, that is batch jobs or TSO users, IAM will generally only hold the record lock(s) from the time of the GET for UPDATE until the record is actually updated or erased, or for records being added, only for the duration of the actual add processing. The exception to this is if IAM RLS is journaling before images of records, implying that a back out could be performed if there is a failure, then the record lock(s) will be held until the program either calls the IAM batch syncpoint, or the job step terminates. If a job step abends while processing a recoverable file, then any record locks obtained for that job step will be retained until a recovery takes place. If there were any other jobs or CICS transactions waiting for the record locks that are being retained until recovery, those requests will be failed when the abend has been detected. The recovery can be performed by IAM's Dynamic Job Back Out function, by IAMJREST, or by whatever other procedures or recovery software that may be available. Information about any retained locks can be found by using the DISPLAY,RETAINEDLOCKS modify command to IAMRLS. Such retained locks will be released upon successful recovery by IAMJREST or Dynamic Job Back Out. The retained locks can also be released by the RELEASELOCKS command.

JOURNALING In addition to handling the I/O and locking services, the IAM RLS address space can also be used for journaling before and / or after images of updated records. The journaling can be performed either to standard DASD sequential datasets, or can be done using the z/OS System Logger. The IAM journaling services are primarily provided to allow for the back out of updates performed by failing batch job(s), or to perform a forward recovery of updates if a file has encountered media damage. CICS will handle it's own transaction back out and other recovery as it does today, using it's own logging mechanisms that is independent of the journaling provided by IAM. By default, IAM RLS will not journal before images for records being updated by CICS transactions because that is done by CICS. If customers want IAM to journal the before images from CICS transactions, they can use the IAM RLS startup parameter CICSJOURNAL.

Information and instructions on setting up IAM RLS journaling is presented in Section 20.30 of the IAM V8.0 Manual.

DYNAMIC JOB BACK OUT An optional Dynamic Job Back Out function for IAM files opened through IAM RLS is available. Whenever a job step abends, all updates done by that job step can be automatically removed by Dynamic Job Backout. Control of Dynamic Job Back Out is provided through IAM RLS DJB startup parameter. If permitted by the startup parameter, DJB can also be controlled via an IAM ACCESS Override. If the batch job step or TSO user has taken IAM syncpoints, then the back out will be performed to the most recent syncpoint taken by the job step prior to abending. If no IAM syncpoints have been taken, then all of the updates performed by the failing job step will be removed on files accessed through IAM RLS. Upon successful completion of the Dynamic Job Backout, all of the retained locks for the failing job will be released.

SECURITY The IAMRLS address space will have to be given security authority to update those files that are going to be processed by IAM record level sharing. IAM does issue the RACROUTE macro within the individual job's address space to validate that the requesting user does have authority to read or update, as appropriate, the IAM dataset being opened prior to requesting that IAMRLS open the dataset. If the RACROUTE indicates that the user / job does not have authority to access the dataset, then the OPEN request is failed. The failing job will receive an IAMW18 error message.

20.01 CONTINUED . . .

RELIABILITY IAMRLS utilizes the various z/OS error handling and recovery facilities to recover from errors and abend conditions that may occur. Our two goals in providing error recovery routines are to provide continuous availability of the services being provided by the IAMRLS address space, and secondly to automatically collect enough information about any failures such that problem determination and correction can be performed from the single failure. The job that had submitted an I/O request to IAMRLS that has experienced a failure will generally be failed with a VSAM logical error code. It will be up to the program to decide whether it can continue processing.

The error data collection will include various messages to the job log, system log, and to the RLS log indicating the abend code, general registers, access registers if applicable, and the failing module. The error routines do attempt to not duplicate diagnostic information that is produced by z/OS, but rather provide additional diagnostic information to be combined with the information provided by z/OS. The error information contained in these messages may be sufficient for problem determination and resolution, particularly if a problem had been previously reported to Innovation. If the error occurred within the IAM RLS address space, an error trace table will be kept in storage for reference, particularly in situations where multiple errors have occurred. Most error situations will also result in a request for a dump to be taken to a system dump dataset, which may include both the IAM RLS address space and possibly the address space of the job that submitted the failing request. The IAM index space associated with the IAMRLS address space may also be dumped.

SERVICEABILITY To aid in serviceability a mechanism is being provided for system support personnel to apply critical fixes to the IAM modules within the IAMRLS address space, without the need to shut down the address space. This facility can also be used to back out fixes should they cause other problems.

20.02 IAM RLS IMPLEMENTATION

This section provides the outline of the procedure to follow to activate and implement IAM RLS. Innovation recommends that you start out with a very limited implementation, and then as you become comfortable with IAM RLS, the use of it can be expanded to more datasets. Many of the parameters that control IAM RLS can be easily modified while IAM RLS is active, which will minimize the need to deactivate IAM RLS to broaden the implementation to more datasets. Within any CICS region or batch job, there can be some IAM datasets using IAM RLS, while other IAM datasets are not using the IAM RLS services.

Careful planning and consideration must be given to how to recover data when a software or hardware error occurs. This is particularly true if you've never used any type of data sharing software in the past. For example, if a batch job abends, you may not be able just to restore a dataset and restart the job because of the potential loss of data from CICS users concurrently updating the dataset. IAM does provide journaling and recovery capabilities that can be used to perform various types of recoveries. You should determine ahead of time what your needs are, which can be different for your different datasets and applications. Careful planning and testing of recovery procedures will be of great benefit to successfully implement IAM RLS.

The various tasks necessary to implement IAM RLS are shown below. Please be sure to review all of the sections to make sure that nothing is missed as you develop your implementation plan.

- Set up the IAM RLS proc within the appropriate system proclibs, and allocate any required parameter libraries. Refer to section 90.10 for the details on the IAM RLS proc and address space requirements.
- Review and select the appropriate values for the various IAM RLS parameters. Refer to section 20.10 for detailed information on the IAM RLS parameters.
- Set up the appropriate criteria for your installation for the IAM RLS automatic dataset selection. Refer to section 20.20 for the details on automatic dataset selection.
- Allocate the IAM RLS log datasets required to support the journaling that you need IAM RLS to perform. Set up the procedures to offload the log datasets, and retain the data for subsequent recoveries as might be needed. Refer to section 20.30 for information on IAM RLS Journaling.
- Review section 20.50 on CICS Transaction Server considerations. Install and activate the necessary CICS exits provided by IAM in every region that will be using IAM RLS. The CICS exits are critical for IAM to perform record locking for CICS. Failure to properly install and implement them will result in record lockouts and other failures. There are no application or JCL changes required for CICS to use IAM RLS.
- Review section 20.60 on batch application considerations. For batch jobs that update a large portion of the records in any recoverable file that is processed by IAM RLS, seriously consider using the IAM RLS Batch Syncpoint program, along with providing for recovery and restart should such programs experience software or hardware failures.
- Plan and develop procedures for data recovery, to be prepared in case of unexpected software or hardware errors occur. Make sure that your implementation for IAM RLS journaling will provide the necessary data to perform data recoveries that will be necessary when failures occur. Refer to section 20.70 for information on recovery of data for IAM files being processed under IAM RLS.
- Review section 20.40 on the IAM RLS operator commands, for information on the various operational requests that can be made of IAM RLS.

20.10 IAM RLS PARAMETERS

There are three sources of parameters for controlling IAM RLS usage. These include the IAM RLS start up parameters, specified by a parmlib mechanism on the IPARMLIB DD statement, the IAM Overrides, and the IAM Global Options Table. The IAM RLS parmlib parameters are completely discussed in this section. There is also a discussion of relevant IAM Overrides with IAM RLS in this section, the full IAM Override documentation is provided in section 30 of this manual. The main global option affecting IAM RLS has to do with specification of automatic eligibility for IAM RLS processing, which is discussed in section 20.20.

IAM RLS PARMLIB

The parameters that can be specified for the IAMRLS address space are presented below. These parameters are passed to IAMRLS via a parmlib mechanism, which is specified via the IPARMLIB DD statement. If you use the example IAMRLS procedure, then the IAMRLS parmlib member is specified using the MBR= operand. You will need a different parmlib member for each unique system on which IAMRLS is being run to specify different log datasets.

The parmlib is read to establish the parameter settings during IAM RLS initialization. Additionally, customers can issue the “CHANGEPARM” command, which will cause an active IAM RLS address space to reread the parmlib member, and cause an immediate update for many of the values. Any special considerations with changing the various parameter values will be discussed in the description of the individual parameters below.

The parmlib dataset must be an 80-column card image dataset, with parameters and values coded in columns 1 through 71. Each card image can contain a single parameter with its values. Multiple parameters can be specified on a single card, but must be separated by a comma and no spaces.

Comment cards can be included, and are indicated by an “*” in the first column of the card. Examples of parmlib are provided in the ICL (Installation Control Library) that was copied from the installation tape. Customers may use the ICL for the parmlib dataset, if so desired.

20.10 CONTINUED . . .

IAM RLS PARAMETERS

Parameter	Description
CICSJOURNAL=	Specifies whether IAM should journal the before images for updates performed by CICS transactions. Valid values are YES or NO. The default value is NO, because CICS normally handles transaction backout and before image journaling itself.
CICSLOCK=	Specifies the action that IAM RLS will take when record lock contention is encountered for a CICS transaction. Three sub-parameters can be provided. The first instructs IAM RLS what action to take when the owner of the lock is another CICS transaction. The second instructs IAM RLS what action to take when the owner of the lock is other than a CICS transaction, such as a batch job or TSO user. The third tells IAM RLS that if it is going to fail the request due to lock contention, it is just to pass a logical error code back to the caller, or abend the transaction. One of the following keywords can be provided to indicate what action IAM RLS will take when the owner of the record lock is a CICS transaction: **CICSWAIT** specifies that the requestor of the lock will wait until the record lock is available when the owner is a CICS transaction. **CICSTIMEOUT** specifies that the requestor of the lock will wait for a short time as specified by the CICSTIMEOUT parameter (see below), if the owner of the lock is a CICS transaction. If the lock is still not available, then the request will fail. **CICSNOWAIT** specifies that the request for the record lock will be failed if the owner is a CICS transaction. One of the following keywords can be provided to indicate what action IAM RLS will take when the owner of the record lock is other than a CICS transaction, such as a batch job or TSO user: **WAIT** specifies that the requestor of the lock will wait until the record lock is available when the owner is a not a CICS transaction. **TIMEOUT** specifies that the requestor of the lock will wait for a short time as specified by the CICSTIMEOUT parameter (see below). If the lock is still not available, then the request will fail. **NOWAIT** specifies that the request for the record lock will be immediately failed when the owner is not a CICS transaction. One of the following keywords can be provided to indicate whether IAM should fail the CICS transaction with an ABEND, or just fail the request following standard VSAM protocol with a logical error code: **ABEND** specifies that IAM will abend the CICS transaction. This requires the installation and activation of an IAM provided CICS Global User Exit, as described later. The transaction will abend with a code of AKC3. **NOABEND** specifies that IAM will fail the request with a logical error code. The default values are: CICSLOCK=(CICSWAIT,TIMEOUT,NOABEND)
CICSTIMEOUT=	Specifies the maximum time, in seconds that a CICS transaction will wait for a record lock if CICSTIMEOUT or TIMEOUT has been specified for the CICSLOCK parameter. Default is 30 seconds.

20.10 CONTINUED . . .

DJB= This parameter provides a way to control Dynamic Job Backout. The following values are accepted:

YES – Specifies that Dynamic Job Backout will be automatically invoked for any job steps that abend, and have recoverable IAM files accessed through IAM RLS. Changes made to recoverable IAM files for the abending job step will be removed from the affected files, back to either the most recent batch syncpoint taken by the job step, or if no syncpoints were taken, then to the beginning of the job step.

NO – Specifies that Dynamic Job Backout will not be automatically invoked, unless the failing job step has opened a recoverable IAM file through IAM RLS with the DJB=YES IAM override.

DISABLED – Dynamic Job Backout support is disabled, and will not be invoked even with a DJB=YES IAM override.

The default values is NO.

JRNPROC= Optional parameter that specifies the name of a procedure (proc) in the system proclibs which IAMRLS will invoke via a START command whenever a log dataset is filled. The name of the log dataset will be passed via the LOG= parameter on the start command. Use of this parameter will aid in automatically backing up and emptying the log datasets in a timely manner, to avoid potential delays should all of the log datasets be filled up.

The default value is that no procedure will be automatically started by IAMRLS when a log fills up, only a WTO message will be issued. Refer to sections 20.30 IAM RLS Journaling and 20.70 IAM RLS Recovery for further information and recommendations regarding this parameter.

20.10 CONTINUED . . .

LOCK= Specifies the action that IAM RLS will take when record lock contention is encountered for a requestor other than a CICS transaction, such as a batch job or TSO user. Three sub-parameters can be provided. The first instructs IAM RLS what action to take when the owner of the lock is another CICS transaction. The second instructs IAM RLS what action to take when the owner of the lock is other than a CICS transaction, such as a batch job or TSO user. The third tells IAM RLS that if it is going to fail the request due to lock contention is it to pass a logical error code back to the caller, or abend the job or TSO user.

One of the following keywords can be provided to indicate what action IAM RLS will take when the owner of the record lock is a CICS transaction:

CICSWAIT specifies that the requestor of the lock will wait until the record lock is available when the owner is a CICS transaction.

CICSTIMEOUT specifies that the requestor of the lock will wait for a short time as specified by the LOCKTIMEOUT parameter (see below), if the owner of the lock is a CICS transaction. If the lock is still not available, then the request will fail.

CICSNOWAIT specifies that the request for the record lock will be failed if the owner is a CICS transaction.

One of the following keywords can be provided to indicate what action IAM RLS will take when the owner of the record lock is other than a CICS transaction, such as a batch job or TSO user:

WAIT specifies that the requestor of the lock will wait until the record lock is available when the owner is a not a CICS transaction.

TIMEOUT specifies that the requestor of the lock will wait for a short time as specified by the LOCKTIMEOUT parameter (see below). If the lock is still not available, then the request will fail.

NOWAIT specifies that the request for the record lock will be immediately failed when the owner is not a CICS transaction.

One of the following keywords can be provided to indicate whether IAM should fail the requestor with an ABEND, or just fail the request following standard VSAM protocol with a logical error code:

ABEND specifies that IAM will abend the batch job or TSO user. The abend code will be a U0185.

NOABEND specifies that IAM will fail the request with a logical error code.

The default values are: LOCK=(CICSWAIT,TIMEOUT,NOABEND)

LOCKTIMEOUT= Specifies the time, in seconds, that a batch job or TSO user will wait for a record lock to become available if the CICSTIMEOUT and / or TIMEOUT value have been specified for the LOCK= parameter. The default value is 30.

LOGDSN1 =
LOGDSN2=
LOGDSN3=
LOGDSN4=
LOGDSN5=
LOGDSN6=

Specifies the dataset name(s) of up to six log datasets. This is required if journaling is going to be used for any IAM dataset under IAMRLS control, and you are not using the System Logger. These datasets are sequential DASD datasets. IAM RLS will determine at startup time which journal was last used, and resume journaling with that journal dataset. It will then subsequently rotate through the list of journal datasets as each journal is filled. If the next journal dataset has not been emptied, or flagged as reusable by IAMJUTIL, IAM RLS will issue a WTOR for the operator to reply to reuse the journal dataset, or to quiesce IAM RLS processing. Additional information on journaling under IAM RLS is in section 20.30.

20.10 CONTINUED . . .

LOGSTRM= Specifies the name of the IAM log stream to be used for journaling, when the System Logger is going to be used. This value must match the log stream name defined by IXCMAIPU. This value is only used if the SYSLOGGER=YES parameter has been specified. The default value, if not specified, is IAMLOGR.sysname, where sysname is the value provided at IPL time in the IEASYSxx member for SYSNAME. Using the system logger is not recommended unless your installation has experience with using the system logger for other purposes, and is familiar with how to use it.

MAXBUFNO= Specifies a value to be used as the default MAXBUFNO for IAM files opened under IAMRLS. If not specified, the default will be the normal IAM default from the IAM Global Options Table. Valid values are from 1 to 2048. This value may be overridden for specific files by an IAM ACCESS override cards that have been included with IAMRLS. If this value is changed when the parameters are reread by the CHANGEPARM command, it will take effect for datasets opened subsequent to the change. Currently opened datasets will not have their MAXBUFNO value changed.

MAXIOTASK= Specifies the maximum number of file I/O tasks to be permitted. This represents the maximum number of concurrent I/O requests that the IAMRLS address space will be able to be actively processing at any point in time. Additional I/O requests will be delayed pending availability of an I/O task to process it. Values from 8 to 255 can be specified. Default value is 64.

MAXJOBS= Specifies the maximum number of jobs that can have IAM files opened under the IAMRLS address space. This is primarily used for setting up the record locking tables. Specifying too small of a value may cause increased overhead for the IAM lock manager if the actual number of jobs exceeds this limit. Values from 1 to 9,999,999 can be specified. Default value is 1024. Changing this value will only take effect when IAM RLS is restarted.

MAXLOCKS= Specifies the approximate maximum number of record locks that are anticipated to be needed at any particular point in time. This value will be used to determine the size of the lock lookup hash table. Specifying too small of a value may cause increased overhead for the IAM lock manager. Values from 1 to 2129903 can be specified. Default value is 133,103. Changing this value will only take effect when IAM RLS is restarted.

MAXTRANS= Specifies the approximate maximum number of concurrently active transactions that a CICS system can have. Too small a value may cause increased overhead for the IAM lock manager. Values from 1 to 9,999,999 can be specified. Default value is 512. Changing this value through the CHANGEPARM command will take effect for CICS regions that are started subsequent to the change.

MESSAGES= Specifies what messages should be printed to the IAMRLS RLSLOGDD file. Valid values are ALL, INFO, or ERROR. ALL is primarily intended for problem diagnosis, as various messages will be written indicating many frequent events. INFO will provide a detailed activity log, including when datasets are opened and closed. ERROR will limit the RLSLOGDD messages to only those issued in error situations. This value can be dynamically changed via operator command. Default value is INFO.

MINBUFNO= Specifies a default minimum number of buffers per IAM file. Valid values are from 1 to 2048. The default value is 1, unless otherwise overridden by an IAMOVRID specification. If this value is changed when the parameters are reread by the CHANGEPARM command, it will take effect for datasets opened subsequent to the change. Currently opened datasets will not have their MINBUFNO value changed.

20.10 CONTINUED . . .

MINIOTASK= Specifies the minimum, and also starting, number of tasks in the IAMRLS address space that will be used to process I/O requests. Values from 8 to 255 can be specified. Default value is 8.

SYSLOGGER= Specifies either YES or NO, to indicate whether the z/OS (or OS/390) System Logger is to be used instead of the sequential datasets. Refer to the section 20.30 on journaling for more information on using this capability. If YES is specified, you must also specify the LOGSTRM= parameter, indicating the name of the log stream. When specified as YES, all of the LOGDSNx parameters are ignored. Default value is NO.

EXAMPLE OF IAM PARMLIB PARAMETERS

An example of parmlib input is shown below:

```
MINBUFNO=20,MAXBUFNO=100
LOGDSN1=MY.LOGDSN1.JOURNAL
LOGDSN2=MY.LOGDSN2.JOURNAL
LOGDSN3=MY.LOGDSN3.JOURNAL
LOGDSN4=MY.LOGDSN4.JOURNAL
LOGDSN5=MY.LOGDSN5.JOURNAL
LOGDSN6=MY.LOGDSN6.JOURNAL
SYSLOGGER=NO
MESSAGES=INFO
CICSLOCK=(CICSWAIT,TIMEOUT,NOABEND)
LOCK=(CICSWAIT,WAIT,ABEND)
CICSTIMEOUT=10
LOCKTIMEOUT=1800
```

Figure 1: Example of IAM RLS Parmlib Parameters

ABENDING CICS TRANSACTIONS

If ABEND was specified for the CICSLOCK parameter, then to cause the abends to occur the IAM provided CICS Global User Exit (GLUE) must be activated. This IAM provided Global User Exit (GLUE) will purge transactions that receive an indication from IAM RLS to do so. This indication will only be presented when the action specified for the IAM RLS **CICSLOCK** parameter is set to **ABEND**. The transaction will abend with a code of **AKC3.** For information on setting this up, refer to section 20.50, CICS Considerations.

IAM OVERRIDES

IAM ACCESS Overrides can be specified for the IAM RLS address space. The overrides can be a member of a card image PDS, such as the IAM ICL. Some of the IAM ACCESS overrides that may be useful to IAM RLS are JRNAD, MINBUFNO, MAXBUFNO and REREAD. The IAM CREATE overrides are not relevant to IAM RLS because no file creations will take place under the IAM RLS address space.

The MINBUFNO and MAXBUFNO can be used on a dataset by dataset basis, to either use more or less buffering than what is being defaulted to under IAM RLS. The need to alter the buffering values should be minimal. This is because IAM RLS offers a MAXBUFNO value to be specified as a default for all files opened under IAM RLS as a start up parameter that was previously described.

The JRNAD override can be used to enable journaling for all IAM files accessed under IAM RLS, by providing an ACCESS DD=&ALLDD,JRNAD=BOTH. Be sure to specify JRNAD on any additional ACCESS override cards provided for specific datasets, as they will not pick up the value from the &ALLDD override. Using the override may be the preferred way to control journaling for IAM RLS, rather than on an individual file basis. The journaling can be turned off for specific datasets by an additional ACCESS override indicating the dataset name.

If you are using the IAM overrides with IAM RLS, then you will most likely want to specify the REREAD keyword on each of the override cards. Specification of REREAD will cause IAM to read the IAM overrides every time an IAM file is opened, rather then only for the first dataset opened, and saving the results in storage. While this is additional overhead, it will allow you to change the overrides without having to stop and restart IAM RLS for the new overrides to take effect.

20.20 IAM RLS AUTOMATIC DATASET ELIGIBILITY

IAMRLS DATASET SELECTION

IAMRLS can automatically decide during OPEN processing which IAM datasets are to be handled by IAMRLS. Automatic selection is based on either the share options that a dataset was defined with, and/or by the dataset name table. An installation specifies the criteria IAM is to use through the IAM Global Options Table, via the RLS option. Using the RLS option, users can specify the minimum share option required for eligibility for IAMRLS processing, and whether or not a dataset name table has been provided to IAMRLS. If a combination of share option and dataset name eligibility is requested, the user can specify via the RLS option whether a dataset must meet both the share option criteria and be in the dataset name table, or if a match on either one sets the dataset as eligible for IAMRLS. The values that can be specified for the RLS Global Option are shown below:

RLS GLOBAL OPTION

RLS Option Value	Description
AND	Indicate that to be automatically eligible for IAMRLS, a dataset must have the specified share option, must not be in the dataset name exclude list, and must be in the dataset name select list.
NONE	Indicates that no datasets are to be considered for automatic eligibility for IAMRLS. Set this option if you are not going to be activating the IAMRLS address space, or if you want to manually direct activity to IAMRLS through the IAM Overrides.
OR	Indicates that to be automatically eligible for IAMRLS, a dataset must have either the specified share option or be in the dataset name select list. If a dataset meets the share option eligibility, then the dataset name include or exclude lists will not be examined. If a dataset does not meet the share option eligibility criteria, then it must be not in the exclude list, and must be in the include list to be selected for IAM RLS processing. OR is the default if both TABLE and a SHAREx value are specified.
SHARE1	Indicates that datasets with any cross-region share option value (1, 2, 3, or 4) will be eligible for IAMRLS processing. Use of this value is not recommended.
SHARE2	Indicates that datasets with cross-region share options of 2, 3, or 4 will be eligible for IAMRLS processing. Use of this value is not recommended.
SHARE3	Indicates that datasets with cross-region share options of 3 or 4 will be eligible for IAMRLS processing. This is the default option as shipped.
SHARE4	Indicates that only datasets with a cross-region share option of 4 will be eligible for IAMRLS processing.
TABLE	Indicates that IAM is to search the dataset name include and exclude table to determine eligibility for IAMRLS processing, subject to other criteria. The default is that the dataset name tables will not take part in eligibility selection for IAMRLS.

20.20 CONTINUED . . .

USING THE RLS GLOBAL OPTION

The discussion below applies to the use of the RLS Global Option. Full use of the IAM Global Option Change Facility is described in Section 91 of the IAM Users Manual. Multiple values can be specified for the RLS option, by enclosing them within parenthesis. When the RLS option is specified on the ZAP control statement, the value(s) specified will act as a complete replacement for the existing RLS values, they do not act in combination with the prior values. As shipped, the default value is **`RLS=SHARE3`**, meaning that IAM files defined with share options of 3 or 4 are eligible for IAMRLS processing. If you subsequently specify **`RLS=TABLE`**, then the share options are no longer considered as criteria for IAMRLS eligibility, only the dataset name inclusion and exclusion tables are considered.

You might desire some combination, for example, lets say that you want all share option 3 or 4 files to be eligible, and some selected share option 2 files. For this, you would specify **`RLS=(SHARE3,OR,TABLE)`**. You would then specify the name(s) of the other files to be included or excluded in the dataset name tables. The reason for both an include and an exclude list is so that you could say, for example, that all datasets beginning with PROD.SHARE should be included for IAMRLS processing, so you would have specified in the dataset name table SELECT DSN=PROD.SHARE* (ending with the single *). However, perhaps there is a file that with that name prefix that you want excluded from IAMRLS, so you could for example specify EXCLUDE DSN=PROD.SHARE.FILE1.

Another possibility is that you might want only share option 3 or 4 files to be eligible for IAMRLS, however you might want to exclude some selected files. In this situation, you would specify **`RLS=(SHARE3,AND,TABLE)`**. Then in the dataset name table, you would specify SELECT DSN=* which will include all datasets, and then exclude specific datasets, such as EXCLUDE DSN=PROD.DONOT.SHARE.FILE1.

RLS DSN TABLE

The dataset name table is provided to the IAMRLS address space through the **IAMDSNTB** DD statement. This DD must specify a card image dataset, either sequential or a PDS member. The table is read in during initialization of IAMRLS, and stored in ECSA. The table is kept in common to minimize the overhead of searching the table during IAM OPEN processing. Customers can update the dataset name tables within the IAMDSNTB dataset, and cause the updated tables to be used by using the **CHANGEDSNT** command of the IAMRLS address space.

Within the IAMDSNTB dataset, two lists of dataset names can be provided. Each dataset name entry is considered a full dataset name, unless it ends with the asterisk (*) character. When the name is suffixed with an asterisk it is treated as a dataset name prefix, meaning any dataset that begins the characters up to the asterisk are considered eligible. A value of DSN=* is allowed, and will be satisfied by any dataset name.

The first list is the dataset name selection list. It is specified by "**SELECT DSN=(**" followed by a list of dataset names, separated by commas. Multiple cards can be used, however each dataset name or prefix must be fully contained within a single card image. A closing parenthesis is used at the end of the last dataset name. If there is just one dataset name, or only an *, then no parenthesis are needed.

The exclude list is specified by the "**EXCLUDE DSN=(**" followed by a list of dataset names, separated by commas. Multiple cards can be used, however each dataset name or prefix must be fully contained within a single card image. A closing parenthesis is used at the end of the last dataset name. If there is just one dataset name, or only an *, then no parenthesis are needed. The EXCLUDE statement must start on a new card.

While each SELECT or EXCLUDE can include a list of dataset names, there can be only one SELECT DSN= statement, and only one EXCLUDE DSN= statement. If multiple SELECT or EXCLUDE statements are provided, only the data from the last one will be saved. All preceding names will be ignored. A maximum of 32,767 names can be specified for each list. However, it is highly recommended that you keep the lists as short as possible, to keep the CSA storage requirement low, and to minimize the searching CPU time. For additional information on acceptable formats, refer to Section 00.03 of the IAM Users Manual for general information on control statement format.

20.20 CONTINUED . . .

RLS DSN TABLE EXAMPLE

Shown below is an example of an RLS DataSet Name table:

```
SELECT DSN=(PROD.PAYROLL.IAM.*,
            PROD.AR.IAM.*,
            IAMV.RLS*,
            PROD.APAY.IAM.*)
EXCLUDE DSN=PROD.PAYROLL.IAM.TESTFILE,
            PROD.AR.IAM.HISTORY,
            IAMV.RLS999.DONOT.SHARE.*)
```

Figure 2: Example of IAM RLS DataSet Name Table

AUTOMATIC RLS DATASET SELECTION

During OPEN processing of any loaded enhanced format IAM dataset, the RLS Global Option is checked. If RLS=NONE is in effect, then normal open processing continues. If the RLS Global Option so indicates, the share options are checked. Then based on the RLS Global Options, if the criteria includes the dataset name tables, the dataset name tables are searched for a match. When IAM refers to the dataset name tables during open processing, first the exclude list is searched. If a matching name or prefix is found in the exclude table, the IAM dataset is treated as being ineligible for IAMRLS processing. If no match was found in the exclude table, then the selection table is searched. If the dataset name or prefix is found in the select table, then the dataset is eligible for IAMRLS processing.

For the preliminary testing phase of IAMRLS, the installation will most likely set a very limited scope. It may be useful to even set the RLS option to a value of NONE, and use the IAM RLS override to initiate processing, or provide a very limited list of eligible datasets.

Subsequent to testing, a wider scope is recommended. Datasets that have shared update access by multiple jobs should be handled by the IAMRLS address space. Such datasets will normally be defined with share options of (3,3), (3,4), (4,3) or (4,4). For that reason, the default selection criteria is for datasets defined with cross-region share options of 3 or 4. Those datasets will be automatically routed to the IAMRLS address space, if it is active.

In some cases, it may also be beneficial for Share Option 2 datasets to be processed by IAMRLS. This would be particularly true if an application requires full read integrity while other jobs or CICS regions are updating the dataset. If there are only a few share option 2 datasets meeting those criteria, then using the dataset name tables would be recommended. However, if almost all of the share option 2 datasets need full read integrity, then it would most likely be better to set the RLS global option to SHARE2.

IAM RLS does incur additional resources, primarily CPU time, to process an I/O request than without IAM RLS. It takes additional resources to ship an I/O request over to another address space for processing, and then return data and status back to the originating address space. For this reason, it is recommended that you only put the IAM files that really need sharing support under IAM RLS, and let standard IAM process the datasets that do not have update sharing active.

20.30 IAM RLS JOURNALING

OVERVIEW IAMRLS provides an optional facility to journal before and / or after images of file updates to IAM files being processed by IAMRLS. These journals can be used to aid in data recovery due to either hardware or software failures. For example, if a media failure occurs and an IAM dataset has to be restored, the IAM RLS journals can be used to reapply the updates that had been made to the file since it was backed up to bring the file status up to the point of failure. If before images are captured then IAM can backout updates performed by failing job steps. The IAM journaling is not a replacement for CICS journaling and CICS dynamic transaction backout, but rather an additional facility that provides for increased recovery capabilities. IAM RLS as a default, will not journal the before images for updates being done by CICS transactions, because CICS will handle the transaction back outs. However, this can be changed if an installation needs to have those records included in the IAM journals.

IAM provides programs to perform recoveries, IAMJREST and IAMBREST. IAMJREST runs as a batch job that can perform forward or backout recoveries. IAMBREST provides for dynamic job step backout of updates performed by a batch job step that abends. IAM also provides a program to perform utility functions on the IAMRLS journals, called IAMJUTIL. IAMJREST and IAMJUTIL can also be used to process journals created by IAM outside of IAMRLS control, as described in Section 10.88 IAM Journal and Recovery. Customers can provide their own utility program to process the data contained in the journals, and to even perform recoveries if so desired. For information on the format of the data contained in the IAMRLS journal records, contact Innovation.

Journaling can be done to either sequential datasets, or using the MVS system logger. Innovation recommends that customers use the sequential datasets for journaling, unless they have experience in using the MVS System Logger facilities. The type of journaling and the journal files are specified by the IAMRLS startup parameters **LOGDSNn, SYSLOGGER, and LOGSTRM**, which are described in Section 20.10.

Customers select the IAM datasets that they want to have IAM journal, by one of the following methods:

- Specification of the IAM ACCESS Override JRNAD on the IAMRLS startup procedure, for example: ACCESS DD=&ALLDD,JRNAD=BOTH.
- Specification of the IAM CREATE Override JRNAD when defining the IAM dataset(s) that are to have journaling.
- Specification of the LOG parameter on the IDCAMS DEFINE for DFSMS managed IAM datasets.

20.30 CONTINUED . . .

SEQUENTIAL LOG DATASETS

The sequential log datasets must have space allocated prior to starting IAMRLS, and are identified by the LOGDSNn= parameter, where n is a numerical digit from 1 to 6. It is recommended that you allocate at least 2 log datasets, preferably 3 or more. Having several large journal datasets will provide for fast access to journal data for job backout recoveries, and help to make sure that there will always be an available journal dataset for IAM RLS, in case some of the journal offload(s) take longer than anticipated. The amount of DASD space required depends on the update activity and the journaling options specified. Each journal record has a header of 112 bytes that is followed by an image of the data record, and is written out in a standard blocked variable length record format. The log datasets may be allocated with secondary space specified, however it is recommended that each log dataset reside on a single volume.

JCL TO ALLOCATE JOURNALS

```
//ALLOCJRN  EXEC  PGM=IEFBR14
//JRN1       DD   DSN=IAMRLS.CPUA.JOURNAL1,DISP=(,CATLG),
//                UNIT=SYSDA,VOL=SER=JRNVL1,SPACE=(CYL,(1000))
//JRN2       DD   DSN=IAMRLS.CPUA.JOURNAL2,DISP=(,CATLG),
//                UNIT=SYSDA,VOL=SER=JRNVL2,SPACE=(CYL,(1000))
//JRN3       DD   DSN=IAMRLS.CPUA.JOURNAL3,DISP=(,CATLG),
//                UNIT=SYSDA,VOL=SER=JRNVL3,SPACE=(CYL,(1000))
//JRN4       DD   DSN=IAMRLS.CPUA.JOURNAL4,DISP=(,CATLG),
//                UNIT=SYSDA,VOL=SER=JRNVL4,SPACE=(CYL,(1000))
//JRN5       DD   DSN=IAMRLS.CPUA.JOURNAL5,DISP=(,CATLG),
//                UNIT=SYSDA,VOL=SER=JRNVL5,SPACE=(CYL,(1000))
//JRN6       DD   DSN=IAMRLS.CPUA.JOURNAL6,DISP=(,CATLG),
//                UNIT=SYSDA,VOL=SER=JRNVL6,SPACE=(CYL,(1000))
```

Figure 3: Example of JCL to allocate IAM RLS Journals

When IAM RLS is started, the journal subtask will look at all of the specified journal datasets. It will select the one with the most recent data (as determined by timestamps within the journal records). If all of the journal datasets are empty, it will start with LOGDSN1. As each journal dataset is filled, or manually switched with the **JSWITCH** command, IAM RLS will use the next higher LOGDSNn dataset. When the highest one has been used, then IAM RLS will switch back to the first one, LOGDSN1.

20.30 CONTINUED . . .

IAM RLS STARTUP PARAMETERS FOR SEQUENTILAL JOURNALS

```
LOGDSN1=IAMRLS.CPUA.JOURNAL1
LOGDSN2=IAMRLS.CPUA.JOURNAL2
LOGDSN3=IAMRLS.CPUA.JOURNAL3
LOGDSN4=IAMRLS.CPUA.JOURNAL4
LOGDSN5=IAMRLS.CPUA.JOURNAL5
LOGDSN6=IAMRLS.CPUA.JOURNAL6
JRNPROC=IAMJRNP1
SYSLOGGER=NO
```

Figure 4: Example of IAM RLS Startup parameters for Sequential Journals

When IAM RLS performs a journal switch, it will start the procedure identified by JRNPROC to offload the journal data, and either empty the dataset or flag it for reuse. If JRNPROC has not been specified, the customer must make sure that the journal dataset is offloaded and emptied prior to IAM RLS requiring the dataset again. Customer's that have automated operations software can trigger a job or procedure by watching for the IAML0510 message, indicating that a journal dataset has been closed. If a journal switch occurs, and the next journal dataset is not ready to be written to, then IAM RLS will issue a WTOR for operator interaction on how to proceed, which will delay IAM RLS processing.

Data from an IAM RLS journal can be offloaded by using the IAMJUTIL program, by IEBGENER, or with their own utility program. Innovation recommends the use of IAMJUTIL because it offers selection criteria on which journal records are copied, and it can indicate that the journal dataset can be reused, while retaining the data in the journal dataset until it is reused. The advantage of reuse is that it simplifies job backout recoveries, because a lot more of the journal data will be readily available than if the journal had been emptied. If you are planning on using the Dynamic Job Backout function of IAMRLS, then you definitely should be using IAMJUTIL to offload the journal datasets.

One scenario for managing the journals is when a journal is full, to use IAMJUTIL to copy only the after images to a tape, which can be used for subsequent forward recovery if needed. The before images, used for backout recovery will be discarded upon reuse of a journal dataset. By providing a large enough set of journals such that the immediate backout needs can be met, this will reduce the amount of journal data that has to be saved.

An example of a "proc" that can be started by IAMRLS, which will copy the after images to a GDG is:

SEQUENTIAL JOURNAL OFFLOAD PROC

```
//IAMJRNP1 PROC LOG=NULLFILE
//*****************************************************************
//*
//*  COPY AFTER IMAGES FROM IAM JOURNAL TO A GDG, AND
//*  MARK JOURNAL AS ELIGIBLE FOR REUSE BY IAMRLS
//*
//*****************************************************************
//IAMJUTIX EXEC PGM=IAMJUTIL
//SYSUDUMP  DD  SYSOUT=*
//SYSPRINT  DD  SYSOUT=*
//IAMJRNL   DD  DISP=SHR,DSN=&LOG
//JRNLOUT   DD  DISP=(,CATLG),DSN=gdg(+1),UNIT=TAPE,DISP=(,CATLG)
//SYSOUT    DD  SYSOUT=*
//SYSIN     DD DISP=SHR,DSN=IAMSYS.RLS.PARMLIB(COPYAFTR)

Contents of IAMSYS.RLS.PARMLIB(COPYAFTR):

COPY AFTER,REUSE
```

Figure 5: Example of PROC to Offload Journal After Image Records

If you will not be making use of the IAM Dynamic Job Backout capability, then other alternative procedures could be used. For example, once a journal dataset is filled, it could be renamed, and space allocated then for the original dataset name. Or, you could have a secondary set of journals, and use the IAMRLS CHANGEPARM command to switch the journaling to the alternate set.

20.30 CONTINUED . . .

SYSTEM LOGGER

As an alternative to the traditional sequential log datasets, customers can optionally use the MVS System Logger DASD-only logging instead. To use the System Logger you must be running an ESA/390 processor or higher with OS/390 Version 2 Release 4 or higher. IAM uses a DASD-only log stream that is not supported in prior versions of OS/390.

IAMRLS can use the MVS system logger for journaling file updates to allow for back out and forward recovery of IAM files. The MVS system logger is a component of MVS that must be activated and configured. For more information on the MVS system logger see the following publications:

OS/390 MVS Setting Up A Sysplex
OS/390 MVS Initialization and Tuning Reference

You must perform the following tasks in order to set up the IAM System Logger environment. Your MVS systems programmer must do some of these tasks.

1. Make sure you have authorization to the MVS system logger address space and that the MVS system logger (IXGLOGR) is running before you define and use IAM log streams. The section "*Define Authorization to System Logger address space*" in the OS/390 MVS Setting Up a Sysplex manual provides more information about this.
2. You must also have authorization to the MVS IXCMIAPU utility. This utility is used to define, update, and delete entries in the LOGR couple dataset. IXCMIAPU is documented in the OS/390 MVS Setting Up a Sysplex manual in appendix B.
3. Use the IXCL1DSU utility to define and format the LOGR couple dataset. Your installation may have already done this if they are using the MVS system logger for other products. This utility is documented in the OS/390 MVS Setting Up a Sysplex manual.
4. Define the IAM log streams in the LOGR couple dataset using the MVS utility IXCMIAPU. See the section "Add Information about Log Streams and Coupling Facility Structures to the LOGR Policy" in the OS/390 MVS Setting Up a Sysplex manual. The following JCL can be used to set the values required for the IAM log stream using the IXCMIAPU utility.

JCL TO DEFINE IAM LOG STREAM

```
//STEP1     EXEC PGM=IXCMIAPU
//SYSPRINT  DD   SYSOUT=*
//SYSIN     DD   *
       DATA TYPE(LOGR) REPORT(YES)
            DEFINE LOGSTREAM NAME(IAMLOGR.CPUB)
            DESCRIPTION(IAM)
            DASDONLY(YES)
            STG_STORCLAS(TMPDATA)
            LS_STORCLAS(TMPDATA)
            STG_SIZE(4000)
            LS_SIZE(16000)
            HLQ(SYSPXCF)
            LOWOFFLOAD(0)
            HIGHOFFLOAD(60)
            DIAG(YES)
/*
```

20.40 IAM RLS OPERATOR COMMANDS

OPERATOR COMMANDS

Certain aspects of the IAMRLS address space can be controlled by an operator by the z/OS (or OS/390) modify (F) command. Some of the functions include shutting down the IAMRLS address space, causing the IAMRLS address space to schedule a dump to a SYS1.DUMPxx dataset, activating and stopping the IAMRLS trace facility, and applying PTF's to an active IAMRLS address space. The general format of the command is as follows:

F iamrls address space name,command

For example, to dump an IAMRLS address space started with the name of IAMRLS, enter the following:

F IAMRLS,DUMP

The following IAMRLS commands can be used. Please note that some of the commands will prompt you to enter additional information via a WTOR.

Command	Description
APPLY,dsn	This command is used to initiate application of Innovation supplied PTF's to the modules loaded in the IAMRLS address space storage. This facility enables customers to apply most maintenance without stopping and restarting IAM RLS processing. For **dsn**, specify the name of a sequential or partitioned dataset containing the fix to be applied. (For PDS's, include the member name in parenthesis.) For example: **F IAMRLS,APPLY,IAMSYS.RLS.ICL(P800001)**
CHANGEDSNT	This command will update the list of dataset names to be included and/or excluded from IAMRLS processing from the IAMDSNTB DD statement. This provides customers with a mechanism for picking up an updated list without having to stop and restart IAMRLS.
CHANGEPARM	This command will update the current IAMRLS parameters from the IPARMLIB DD statement file. While most parameters can be dynamically changed, some parameters will only effectively be changed by a restart of IAMRLS. For example, because MAXLOCKS is used to set an internal table size established at startup, changing that value will not have any effect until IAMRLS is restarted.
CLOSEFILE	This command will close the specified dataset. Additional operands, separated by commas, are: **DSN=** Specifies the dataset name. One of the following is also required: **JOBNAME=** Specifies the jobname for which the dataset is to be closed. **JOBID=** Specifies the JES JOBID that identifies the job for which the dataset is to be closed. **ALL** Specifies that the file is to be closed for every job that has it opened under IAM RLS.

20.40 CONTINUED . . .

DISPLAY This command, along with associated sub-parameters, will display requested information on the output file RLSLOGDD. The RLSLOGDD is used as a default because the output from the DISPLAY could be quite substantial. By adding the keyword ‘,CON’ to the end of the DISPLAY request, the information will be displayed on the operator’s console. One of the following parameters must be provided to indicate what information is to be displayed.

CONTENTION – Requests IAM RLS to display any record locks held that have contention (other requestors waiting for the lock).

EXCLUDETB – Indicates that the dataset name exclusion table is to be displayed.

RETAINEDLOCKS[,DSN] – Indicates that information on the jobs, and optionally which datasets used by those jobs, have had locks retained due to an abend will be displayed.

SELECTTB – Indicates that the dataset name selection table is to be displayed.

For example: **F IAMRLS,DISPLAY,RETAINEDLOCKS**

DJBTRACE= This command is used to activate or deactivate the Dynamic Job Backout tracing. The Dynamic Job Backout trace is useful for determining exactly what file activity is being performed by Dynamic Job Backout. This can be used to verify proper operation, or for problem diagnosis relating to the recovery processing. When the trace is activated, by specifying DJBTRACE=YES, the next time a backout is attempted, the backout will dynamically allocate the RLSDJBDD file to a SYSOUT data type of dataset. When deactivated, by DJBTRACE=NO, upon completion of a backout, the output for the trace will be closed and deallocated. Valid values are:

YES – Activate IAM RLS Dynamic Job Backout tracing.

NO – Deactivate IAM RLS Dynamic Job Backout tracing.

DUMP This command is used to request that the IAMRLS address space take a dump to a SYS1.DUMPxx dataset. The dump will automatically include the data space containing the IAM Index Space.

JSWITCH This command causes logging to switch from the currently active log dataset to the next log dataset.

MESSAGES= Specifies what messages should be printed to the IAMRLS RLSLOGDD file. Valid values are ALL, INFO, or ERROR. ALL is primarily intended for problem diagnosis, as various messages will be written indicating many frequent events. INFO will provide a fairly detailed activity log, including when datasets are opened and closed. ERROR will limit the RLSLOGDD messages to only those issued in error situations.

QUIESCE[,FORCE] This command requests that the IAMRLS address space shut down. If FORCE is not specified, then when all of the currently open IAM files are closed, the IAMRLS address space will terminate. Any new open requests will be rejected.

If the FORCE keyword is specified, then all of the files will be immediately closed, and the address space will terminate normally. All subsequent requests by the jobs that were processing datasets through the IAMRLS address space will be rejected.

20.40 CONTINUED . . .

RELEASELOCKS
RELLOCKS Specifies that IAM RLS is to release locks for the specified job or CICS transaction. Either retained or held locks may be released. Additional parameters must also be provided to indicate the job step for which locks are to be released. The information necessary to provide the detailed required by this command can be obtained from the DISPLAY,RETAINEDLOCKS command. The following sub-parameters must be specified:

JOBNAME= specifies the 1 to 8 character job name.

JOBID= Specifies the 8 character JES jobid, for example: JOB12345

ASID= Specifies the hexadecimal digits of the address space ID number.

STEPNAME= Specifies the 1 to 8 character step name, and is optional.

TRNAME= Specifies the 1 to 4 character CICS transaction name.

TRID= Specifies the CICS transaction number, in 1 to 5 decimal digits.

RESTART This command will reverse a previously issued QUIESCE command that was issued without the FORCE keyword. Normal processing will be resumed by the IAMRLS address space, and new open requests will be honored. This will only function if the IAMRLS address space has not already terminated.

RESTORE,dsn This command can be used to reverse the application of an Innovation supplied PTF from the modules loaded within the IAM address space. This can be used when a PTF has been applied using the APPLY command, and is either causing additional problems, or if it did not resolve the problem. For **dsn** specify the same value as was provided on the APPLY command.

TRACE This command activates the IAMRLS internal tracing. Refer to the section on IAMRLS tracing for additional information on using the trace facility.

20.41 IAM RLS TRACING

IAMRLS TRACE FACILITY

The IAMRLS Trace Facility enables Innovation support personnel to request information about I/O requests and other processes that are occurring within the IAMRLS Address Space. Trace output will be to dynamically allocated SYSOUT spool datasets or to previously allocated output files. All trace requests are initiated via the MVS console MODIFY (F) command. If additional parameters are required beyond what was initially specified, then the trace processor will request them via a WTOR. For trace start requests, the additional parameters can alternatively come from a sequential dataset. The general format of using the IAMRLS trace command is as follows:

F iamrls,TRACE,subcommand,parameters

The “iamrls” is replaced by the IAM RLS address space name. The “subcommand” are one of the subcommands shown below. The parameters will be as indicated for the subcommand. There are three IAM trace subcommands for IAMRLS and all are entered via the MVS MODIFY command. The subcommands are:

Subcommand	Description
START	Request a new trace
STOP	Stop an existing trace
LIST	List all active trace requests

20.41 CONTINUED . . .

STARTING A TRACE

To start a trace, issue the command:

F iamrls,TRACE,START

The additional parameters can be included on the above command, provided via replies to the WTOR's, or with the PARMS= parameter, through a sequential dataset. Parameters for TRACE START subcommand are:

Parameter:	Description:
DSN=	Optional parameter that specifies the name of the IAM dataset that you wish to trace requests against. Trace information will be generated by any request from any address space that is accessing the named file unless further limited by JOBNAME= and/or STEPNAME= parameters.
END	Required parameter to signify that all of the desired parameters had been specified.
ID=	Optional parameter that specifies a unique 1 to 8 alphanumeric character string used to identify this trace request. If one is not specified, IAM will generate a value. This ID is required for STOP and LIST requests.
JOBNAME= **JOB=**	Optional parameter that specifies tracing is to be performed for requests from specific job. If not specified, all jobs with IAM datasets that are accessed through the IAM RLS address space will be eligible for tracing. Use this parameter with DSN= and/or STEPNAME= to limit what is traced.
OUTDSN=	Optional parameter that causes trace output to be sent to an MVS file instead of a dynamically allocated SYSOUT file. If specified, it must be a pre-allocated and cataloged sequential dataset. IAM will set up the required DCB characteristics.
PARMS=	Optional parameter that specifies the name of a sequential file that contains the actual trace request. If not specified, all of the necessary parameters must either be specified on the modify (F) command itself, or provided as replies to the WTOR messages that will be issued.
STEPNAME=	Optional parameter that specifies to trace only requests to IAMRLS from the specific step name. When used without other limiting parameters, this will cause ALL requests from any job that has the specified STEPNAME to be traced. When not specified, all step names are eligible for tracing. Use with DSN= and/or JOBNAME= to limit what is traced.
TRACEREQUEST=	Required parameter that specifies what type of trace records are to be produced. Any or all options may be specified on any trace request. The options are: • **IOS** – Trace I/O Start requests • **IOE** – Trace I/O End requests • **BFR** – Trace IAM Buffer requests • **EXCP** – Trace IAM EXCP requests • **XTND** – Trace IAM extend requests • **PC** – Trace IAMRLS PC calls

20.41 CONTINUED . . .

STOPPING A TRACE

To terminate tracing, issue the following command:

F iamrls,TRACE,STOP,ID=traceid,END

The parameters for TRACE STOP requests are:

Parameter:	Description:
ID=	Required parameter that specifies the ID of the TRACE request to be stopped.
END	Required to signify that all parameters have been specified.

DISPLAYING ACTIVE TRACE REQUESTS

Use the TRACE,LIST command to display the currently active trace requests. You can request either all of the currently active trace requests, or a specific trace request.

The parameters for TRACE,LIST requests are:

Parameter:	Description:
ID=	Optional parameter that specifies the trace ID from the TRACE,START request. If not specified, all active trace requests will be displayed on the MVS console.
END	Required parameter to indicate that all of the desired parameters have been specified.

EXAMPLES OF TRACE COMMANDS

The following demonstrate some examples of using the IAM RLS trace function. All examples assume an IAMRLS address space job name of IAMRLS.

To trace all I/O Start and I/O End requests from all IAM files from CICS region CICDIDP1:

F IAMRLS,TRACE,START,ID=IAM1,JOBNAME=CICSIDP1,TRACEREQ=(IOS,IOE),END

To trace all PC calls against dataset MY.IAMFILE:

F IAMRLS,TRACE,START,ID=PCTRACE,DSN=MY.IAMFILE,TRACEREQ=PC,END

To list all active trace requests:

F IAMRLS,TRACE,LIST,END

To stop a trace request with the ID of TRACE1:

F IAMRLS,TRACE,STOP,ID=TRACE1,END

20.50 IAM RLS CICS CONSIDERATIONS

OVERVIEW

CICS (or Transaction Server) customers can use IAM files under IAM RLS, without changes to their application programs or transactions. Additionally, there are no changes required to the IAM file definitions under CICS. There are a couple of IAM provided CICS exit routines that must be installed and activated to use IAM RLS under CICS. These exits are needed so IAM RLS can provide the required record locking protocols. Exits are used for IAM RLS to recognize when a logical unit of work (UOW) completes due to a SYNCPOINT request, or transaction termination, which is done using the CICS Task Related User Exit (TRUE) facility. IAM also provides a CICS Global User Exit (GLUE) for backout to recognize when a transaction backout is taking place that is subsequent to the original transaction, such as can occur during an emergency restart. If these exits are not activated, then attempts to OPEN IAM files for IAM RLS processing under CICS will be failed.

There is an additional Global User Exit (GLUE) that is provided and must be installed if it is desired that IAM abend any transactions that have encountered lock contention, and have indicated that IAM RLS is to ABEND such transactions.

IAM RLS CICS EXITS

IAM customers that wish to use the IAM RLS functionality in their CICS regions must install and activate the provided Task Related User Exit (TRUE), IAMBCICS, and the provided Global User Exit (GLUE), IAMXFCBO. To do this, the following steps are required.

- Insure that modules IAMXCINI, IAMBCICS, and IAMXFCBO are in a dataset that is part of the DFHRPL concatenation in the CICS JCL.
- Insure that modules IAMXCINI, IAMBCICS and IAMXFCBO are defined to CICS in DFHCSD. Each module must be defined as
  - LANGUAGE=ASSEMBLER
  - RELOAD=NO
  - DATALOCATION=ANY
  - EXECKEY=CICS.
- Add IAMXCINI to the second stage of the PLTPI list that is used at CICS startup. An example of how this might look is:

```
DFHPLT TYPE=INITIAL,SUFFIX=X1
DFHPLT TYPE=ENTRY,PROGRAM=DFHDELIM
DFHPLT TYPE=ENTRY,PROGRAM=IAMXCINI
DFHPLT TYPE=FINAL
END
```

- Restart CICS.

The exit activator program, IAMXCINI, programmatically issues the commands:

1. EXEC CICS ENABLE PROGRAM(IAMXFCBO) EXIT(XFCBOUT) START
2. EXEC CICS ENABLE PROGRAM(IAMBCICS) TASKSTART START

These commands activate the TRUE and the GLUE needed for RLS functionality in the CICS region.

20.50 CONTINUED . . .

GLUE FOR RECORD LOCK TIMEOUT

For some IAM RLS users it may not be practical to deal with the “ILLOGIC” condition raised in CICS when a transaction has timed out while waiting for a record lock. This can occur if a customer has specified that timeouts should occur when waiting for locks under IAM RLS on the CICSLOCK parameter (See section 20.10). For those users, a program is provided that runs as a CICS Global User Exit (GLUE), which will purge transactions that receive the IAM RLS specific ILLOGIC condition.

A site wishing to activate this functionality needs to do the following:

- Insure that modules IAMXFCAB and IAMXFCAT are in a dataset that is part of the DFHRPL concatenation in the CICS JCL.
- Insure that modules IAMXFCAB and IAMXFCAT are defined to CICS in DFHCSD.
- Add IAMXFCAT to the second stage of the PLTPI list that is used at CICS startup.
- Restart CICS.

The exit activator program, IAMXFCAT, programmatically issues the command “EXEC CICS ENABLE PROGRAM(IAMXFCAB) EXIT(XFCREQC) START”. This command causes CICS to enter IAMXFCAB at the conclusion of each file request made through the standard CICS API before returning control to the application logic.

The exit program itself checks the EIBRCODE for the IAM RLS deadlock specific indicators, and if they are found it programmatically issues a command to purge the transaction. This results in a transaction abend with a code of AKC3. Users that decide to use this feature will have to determine if they wish to suppress this dump code or use the dumps to identify the files and the activities that are involved in dead lock situations.

Users that have other program(s) enabled for the XFCREQC exit should generally enable IAMXFCAB last, CICS will run programs at exits in the order they were enabled.

20.60 IAM RLS BATCH CONSIDERATIONS

OVERVIEW

The ability to maintain and achieve data integrity can be a complex matter when dealing with non-transactional processing, as is the case with many batch applications. Such batch applications range from performing a relatively simple and straightforward update process on one or only a few files, to very complex processing involving updates to a large number of related datasets and databases. Additionally, such applications may not have been designed or written with record level data sharing as a consideration. While software products, such as IAM RLS, can support and facilitate data sharing, there may be a need to change some batch applications to facilitate good performance and data availability in a data sharing environment.

The discussion of IAM RLS and batch processing below is intended to provide assistance in determining what, if any, changes may be necessary to the batch job streams that will be processing files managed by IAM RLS. The main considerations that should be kept in mind are what are the recovery requirements for the files being processed by the batch jobs and what other concurrent activity is likely to be occurring on the shared files while the batch job(s) are executing. One of the key points is that with concurrent updating going on, you can not just restore all of the files that were updated by the batch job and rerun it, because updates made by the other concurrent processing will then be lost. Different recovery techniques will have to be used.

IAM RLS AND BATCH JOBS

IAM RLS can be used with batch jobs that read and update IAM files. With IAM RLS automatic dataset selection, there are no changes required to JCL for batch jobs to use IAM files under IAM RLS. The main area that requires some consideration is for those batch jobs that update recoverable IAM files with IAM RLS. For purposes of this discussion, recoverable IAM files means those files that are journaling before images. Because such files can have updates backed out should the batch job fail, the record locks are retained until the batch job step ends normally, until a batch syncpoint call is performed, or if the job step fails, until a recovery is performed.

A job step failure from the IAM RLS viewpoint would be a job step abending. IAM RLS does not look at application return codes, or the condition codes set for a particular job step. If the job step abends, it is considered a failure. If a job step terminates without abending, then IAM RLS considers it successful. This is significant because if a recovery is needed, a valid recovery would consist of only recovering to either the last syncpoint, if any, or to the beginning of the job step if no syncpoints were taken. Attempting to backout updates performed by prior job steps may cause a loss of data, because other jobs or online systems may have updated those records subsequent to their completion. Those updates would be lost if a recovery is performed for job steps other than the failing job step. Likewise, attempting to backout updates for a job step the application considers failed due to condition codes could also result in data loss. Backouts of all of the updates made by the failing job could only be performed if the only update access was being performed by the failing job. This can only be guaranteed if the failing job allocates the IAM dataset(s) with DISP=OLD. Otherwise, there is nothing to prevent the IAM files from being updated by other jobs or online systems.

Batch jobs that only read shared IAM files can be executed under IAM RLS generally without any JCL or programming changes. Batch job steps that run relatively quickly, say in less than 15 to 30 minutes, and that update a relatively small portion of the records in the dataset, say less than 10% should also be able to use IAM RLS recoverable files without any changes.

Batch jobs that update IAM RLS files, which have long run times or update a large portion of the records in recoverable files may need to be changed, particularly if they are running concurrently with online systems or other batch jobs.

20.60 CONTINUED . . .

BATCH SYNCPOINT PROCESSING

IAM RLS has a syncpoint capability that can be used by batch applications. When recoverable files are being updated, the record locks are retained for records that are updated, added, or deleted until the job step terminates. If many records are being updated, then the number of locks held could be quite high, resulting in effectively locking out CICS transactions until the batch job completes. Such lock retention could also result in deadlocks between CICS and the batch job, resulting in failed requests. To prevent these types of problems, customers are encouraged to consider the use of the IAM batch syncpoint for such programs. Customers with other VSAM sharing packages may already have such logic built into their batch programs, and will only need to change the syncpoint program they are calling.

The IAM batch syncpoint service will perform a few different functions. First, it will make sure that all updated blocks for IAM files that are being processed by IAM RLS for this job have been physically written to DASD. If blocks have not yet been written, such output I/O will be performed. Upon completion of any outstanding buffer writes, the batch syncpoint will then write out a syncpoint record in the journal, that can be used if a back out recovery is necessary. Then, any record locks held by that batch job will be released.

IMPLEMENTING BATCH SYNCPOINTS

The amount of effort to implement a syncpoint within a batch job, along with the associated restart processing may be minimal, or may be rather substantial. This all depends on the nature of the updates being performed, the restart capabilities that the application program may already have, and how other updated files are handled. There are certain types of updates for which an application program can simply be restarted, and it can reapply those updates without the need for any recovery, and retain full data integrity. An example of this could be a job that updates addresses of customers. Such updates may not matter if they were already done before, the program will just end up repeating the update with the same data, or the application may have a way to check to see if the record was previously updated, and if so it would not have to reapply the update. For these types of programs, all that would need to be done is to periodically call the IAM batch syncpoint to release the locks of records updated to that point in time.

For other applications, such as those that update account balances, a more complex approach may be necessary. These types of programs will need to implement a more complex syncpoint and restart mechanism, such that the program can be restarted from a specified point in time, and be able to resume processing from that syncpoint. When an application is able to do that, it can issue IAM batch syncpoints. If it needs to be restarted, back outs can be performed to for example the most recent syncpoint and the job will not perform any double updates. The batch application program will in effect, have to perform a processing checkpoint, and include in that the execution of the IAM batch syncpoint. The application will have to implement their own type of checkpoint processing, as MVS Checkpoint cannot be used with IAM files being processed under IAM RLS.

Calling the IAM Batch syncpoint is easy. Just issue a call to IAMSYNC. No parameters are necessary. The program IAMSYNC is a small stub, and designed to be link-edited in with the application program. An alternative is for the application to LOAD the IAMSYNC module, saving the address and calling it as necessary.

20.70 IAM RLS DATA RECOVERY

RECOVERY OVERVIEW

One of the most important considerations when implementing IAM RLS, or any type of data sharing, is how to recover should errors, media failures, or outages occur. The journaling and recovery tools provided with IAM will assist in the recovery of data for the IAM files that are processed under IAM RLS. The intent of the information presented in this section is to assist you in planning for your data recovery needs, and to help insure that appropriate procedures are put in place for the protection of your data.

This section will address performing backout recoveries for failing batch jobs, and forward recoveries from restored IAM datasets. CICS provides the recovery for failed transactions with dynamic transaction backouts, so IAM RLS does not provide that capability.

IAM RLS JOURNALS

In order to perform recoveries with the IAM tools, you must provide for IAM RLS Journaling, as described in Section 20.30. You'll need to decide on what type of recoveries you'll want to be able to perform. If you want to be able to backout updates from failing job steps, you'll need to specify the journaling of before images. If you want to be able to recover files after restoring them from a backup, you'll need to be collecting the after images. To provide the best possible recoverability, Innovation recommends that you journal both before and after images. This way, you'll be able to perform whatever recoveries may be necessary. To insure that journaling is being performed for all of your files, it is recommended that you provide the appropriate ACCESS overrides to the IAM RLS address space that will specify the level of journaling you need. For example, to specify journaling for all datasets, you can use the following IAM override control card for the IAM RLS proc:

```
ACCESS  DD=&ALLDD,JRNAD=BOTH
```

If you have other override cards, be sure to specify the JRNAD=BOTH on those as well, as the DD=&ALLDD override card is effective only for datasets that are not otherwise explicitly specified on override control cards.

20.70 CONTINUED . . .

MANAGING THE IAM RLS JOURNALS

When allocating the journal datasets for IAM RLS, it is recommended that you provide space for at least 24 hours of processing. Providing that amount of journal space across four to six journal datasets will provide the capability for performing backout recoveries within a 24-hour period without having to research what journal datasets need to be used. Innovation also recommends the use of the provided program, IAMJUTIL, with the REUSE option to copy the data from the active journal files to an accumulation dataset. This will keep the journal data within each IAM journal dataset until it is needed for reuse. Also, be sure to provide for an automated mechanism to offload journal data from the IAM RLS journal datasets, to insure that there it will be ready when needed. This can be accomplished with the IAM RLS JRNPROC= parameter, or by using automated operations software that looks for the message that an IAM journal file has been filled, and is being switched to the next journal file.

If you have sufficient DASD space, one method to handle the accumulation of the journal data is to have the procedure that offloads data from an IAM RLS journal to place that data in a staging dataset. If you have adequately sized journals, it will only be necessary to copy those journal records necessary for a forward recovery (i.e. the after images). Each time an IAM RLS journal is filled, the data from the most recently filled journal is added to the end of the staging dataset. Then, on a scheduled periodic basis, such as daily, copy the data from the staging dataset, and put it out to a GDG on tape, then clear out the staging dataset. An example of what your offload proc might look like is shown below:

```
//JRNOFFLD PROD LOG=NULLFILE
//*****************************************************
//*
//*    COPY AFTER IMAGES FROM AN IAM RLS JOURNAL
//*    TO AN ACCUMULATION DATASET.
//*    MARK JOURNAL AS ELGIBLE FOR REUSE
//*
//*****************************************************
//OFFLOAD   EXEC PGM=IAMJUTIL,REGION=6M
//SYSUDUMP  DD   SYSOUT=*
//SYSPRINT  DD   SYSOUT=*
//IAMJRNL   DD   DISP=SHR,DSN=&LOG
//JRNLOUT   DD   DISP=MOD,DSN=PROD.IAMRLS.JOURNAL.ACCUM
//SYSOUT    DD   SYSOUT=*
//SYSIN     DD   DISP=SHR,DSN=IAMSYS.RLS.PARMLIB(OFFLOAD)

Contents of IAMSYS.RLS.PARMLIB(OFFLOAD):

      COPY  AFTER,REUSE
```

Figure 6: Example of IAM RLS Journal Offload to Accumulation DataSet (EX2070A)

20.70 CONTINUED . . .

The job to copy the data from the accumulation file to tape can use either IAMJUTIL or IEBGENER. If you wanted keep the before images in the accumulation dataset, and only copy the after images to tape, you would want to use IAMJUTIL. An example is shown below:

```
//*****************************************************
//*
//*    COPY AFTER IMAGES FROM AN IAM RLS JOURNAL
//*    ACCUMULATION DATASET TO A GDG ON TAPE.
//*    THEN EMPTY THE ACUMMULATION DATASET.
//*
//*****************************************************
//COPYACUM EXEC PGM=IAMJUTIL,REGION=6M
//SYSUDUMP   DD   SYSOUT=*
//SYSPRINT   DD   SYSOUT=*
//IAMJRNL    DD   DISP=OLD,DSN=PROD.IAMRLS.JOURNAL.ACCUM
//JRNLOUT    DD   DISP=(NEW,CATLG),DSN=PROD.IAMRLS.DATA(+1),
//                UNIT=TAPE
//SYSOUT     DD   SYSOUT=*
//SYSIN      DD   *
      COPY   AFTER
/*
//MPTYACUM EXEC PGM=IEBGENER
//SYSUT2     DD   DISP=OLD,DSN=PROD.IAMRLS.JOURNAL.ACCUM
//SYSUT1     DD   DUMMY,DCB=PROD.IAMRLS.JOURNAL.ACCUM
//SYSPRINT   DD   DUMMY
//SYSIN      DD   DUMMY
```

Figure 7: Example of copying journal data to GDG on tape (EX2070B)

DYNAMIC JOB BACKOUT IAM RLS offers an optional capability to dynamically backout changes made to IAM RLS files that were made by an abending job step. The use of Dynamic Job Backout is controlled by the IAM RLS parameter DJB, as discussed in Section 20.10. When a job step that has updated IAM files managed by IAM RLS abends, if Dynamic Job Backout can be used by that job, then IAM RLS will internally schedule the backout. The backout will be performed under the IAM RLS address space, with appropriate messages issued upon completion. Upon completion of the backout, IAM RLS will release any record locks retained by the failing job step. If the abending job step had taken any IAM batch syncpoints, then the backout will be performed to the last syncpoint taken by the job step. If no syncpoints are found, then all updates made to IAM files managed by IAM RLS will be backed out.

MANUAL JOB BACKOUT If you have not enabled Dynamic Job Backout, and a batch job step abends, you can use IAMJREST to perform a backout of the failing job step. You would not normally want to perform a backout of the updates made by the preceding job steps because those record locks would have been released upon normal step termination, and those records could have subsequently been updated by another batch job or online system. An example of a manual job backout is shown below. IAMJREST will automatically release any retained locks upon completion of the backout.

```
//BACKOUT  EXEC  PGM=IAMJREST,REGION=0M
//SYSPRINT DD   SYSOUT=*
//SYSOUT   DD   SYSOUT=*
//IAMINFO  DD   SYSOUT=*
//SORTWK01 DD   UNIT=SYSDA,SPACE=(CYL,(50,10))
//SORTWK02 DD   UNIT=SYSDA,SPACE=(CYL,(50,10))
//SORTWK03 DD   UNIT=SYSDA,SPACE=(CYL,(50,10))
//SYSIN    DD   *
      RESTORE BACKOUT,JOBNAME=jobname,STEPNAME=stepname,
JOBID=JOBnnnnn
/*
```

Figure 8: Example of JCL for manual backout (EX2070C)

20.70 CONTINUED . . .

FORWARD RECOVERY

A forward recovery may be necessary if an IAM file is damaged due to hardware media failures, or due to software failures. The first step for such a recovery is to restore the IAM dataset from a backup copy. The backup could have been done by an FDRREORG, IDCAMS, or some other application, or it could have been done by a DASD management utility such as FDR or DFSMSdss. Once the dataset is restored, you can use IAMJUTIL to reapply all of the updates that had occurred to the dataset prior to the failure. For the forward recovery, you would want to specify the FROMDATE and FROMTIME being the time that the backup started, and specify the IAM dataset that is being recovered.

In the example below, there is the presumption that the current IAM RLS journal datasets have all of the necessary data to perform the recovery, so no journal datasets are specified in the JCL.

```
//RECOVER   EXEC PGM=IAMJREST,REGION=0M
//SYSPRINT  DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
//IAMINFO   DD  SYSOUT=*
//SORTWK01  DD  UNIT=SYSDA,SPACE=(CYL,(50,10))
//SORTWK02  DD  UNIT=SYSDA,SPACE=(CYL,(50,10))
//SORTWK03  DD  UNIT=SYSDA,SPACE=(CYL,(50,10))
//SYSIN     DD  *
      RESTORE FORWARD,DSN=bad.iam.file,FROMDATE=yyyyddd,
      FROMTIME=hhmmss
/*
```

Figure 9: Example of a Forward Recovery (EX2070D)

THIS PAGE WAS INTENTIONALLY LEFT BLANK

30.01 IAM OVERRIDE STATEMENTS OVERVIEW

OVERVIEW IAM offers various features and capabilities that are not available with VSAM. One of the ways to indicate to IAM what features and capabilities are to be used with any specific file is through the IAM Override facility. Many features can be activated or deactivated as defaults through the IAM Global Options facility. Others, such as IAM's Dynamic Tabling, are available only through an override specification. There could also be circumstances where the user does not want the default values from the IAM Global Options Table, or wants to make sure a certain capability is being used. For example, with a very highly accessed IAM file, the user may want to increase the maximum and minimum number of buffers that IAM is to use for processing that file.

This section is intended to provide in one place, a reference for all of the IAM Overrides that are applicable with IAM Version 8.0 using the IAM VSAM (VIF) interface. The various facilities and capabilities of IAM are described in other places of the manual. Descriptions of the override keywords for the older interfaces, i.e. Native or ISAM, along with overrides that are not recommended for use, are contained in the Installation Control Library (ICL) as member OLDOVRID. That library was downloaded as part of the install process. Contact your systems programmer if you need to reference that material.

IAM offers the user two types of override statements, the CREATE override and the ACCESS override.

CREATE STATEMENT The CREATE statement can be used when a file is defined, loaded or reorganized to specify file attributes and special IAM features. Using the CREATE statement, an IAM user can tailor file processing by changing or overriding file characteristics and options that would otherwise be based on an IDCAMS DEFINE and the defaults in IAM's Global Option table. In some cases IAM's default values may not be sufficient. For example, for a file that has very heavy access, which could benefit from using more than the default number of buffers use the CREATE statement to override the defaults for MAXBUFNO and MINBUFNO on the file define or load, which will eliminate the need to override the file by every job that accesses that file.

ACCESS STATEMENT The ACCESS statement can be used when a loaded file is processed for input or update to override IAM run time options for a specific job. Using the ACCESS statement an IAM user, at execution time, can tailor a file's processing. The user can specify options that do not exist in VSAM (ex: Dynamic Tabling of Data Records) or adjust IAM's Real Time Tuning (ex: MAXBUFNO, MINBUFNO). The overrides specified on an ACCESS statement are used only for that particular jobs step, and are not saved with the file as certain CREATE override values are.

IAMOVRID DD CARD The IAMOVRID DD card specifies the Override control data set. This data set is in card image format, and typically an input stream (i.e. DD *) data set. IAMOVRID may be used in any job step that processes IAM files (IDCAMS, CICS, batch applications, etc.). The IAMOVRID file is only read once per job step, unless the REREAD keyword is specified. The file is read when the first IAM file in the job step is opened. A table of the overrides is built in extended private storage, and is referenced for subsequent IAM file opens. The table remains in storage until step termination. Up to two hundred override statements can be provided.

30.01 CONTINUED . . .

OVERRIDE STATEMENT FORMAT

IAM Override statements are contained on one or more 80-character records coded in positions 1 through 71. Each control statement starts with a command (ex: CREATE or ACCESS). The command can begin in the first column, or can be preceded by some blanks. The command is immediately followed by one or more blanks, then one or more optional keywords with their values, separated by commas. The last keyword or keyword/value pair in a statement must be followed by one or more blanks. Anything following the blank will be treated as comments. Comments are not permitted prior to the first keyword in a statement. The entire line will be treated as a comment when a '*' is placed in the first column.

To continue a statement onto an additional line, a comma and at least one blank must follow the last keyword or keyword/value pair on that line. The next keyword is then begun on the following line. When a keyword requires an optional variable, that keyword and its variable must start and end on the same line.

Each override statement indicates the file(s) to which it applies through the DD= or DSN= keyword. Up to 40 different DD names or one data set name (DSN) can be specified for any particular override statement. Additionally, an override command can be made applicable to all IAM files in that job step by coding DD=&ALLDD. Such an override will be effective for all IAM files except those that have their own explicit override. If multiple overrides of the same type (i.e., CREATE or ACCESS) are specified for the same DD name, then the overrides on the last one will be used for that execution, with ALL values from prior cards being replaced. The DD name field has precedence over the data set name (DSN). The file(s) and data sets referenced on the overrides do NOT have to be included in the JCL. IAM data sets can be dynamically allocated. This is frequently the case with CICS systems, and when using IAM Alternate Index support. When an IAM file is opened, IAM scans through the table of overrides to see if there is a match for the DD name of the file being opened. If so, the specified overrides are used. If no matching DD name is found, then IAM scans through the DSN= overrides looking for a match. If none is found, and the &ALLDD override has not been specified, then the file will be processed using defaults from the IAM Global Options Table.

The various override keywords can be abbreviated, if desired. The underscored portion of the keyword indicates the minimum abbreviation allowed when it is described.

OVERRIDE SYNTAX ERRORS

If an error is detected on one or more of the provided override cards, IAM will print the contents of the override card on the job log, along with a message indicating the error. If an IAM files is being defined or loaded and an override error is encountered, then the file definition or load will be failed. Override errors are tolerated for normal file access, with the bad override statement(s) being ignored. The reason a define or file load is failed when there are override errors is that the file may not be defined or loaded with the desired attributes, which may cause processing problems in subsequent programs that use the file. Normal file access overrides, while perhaps critical for performance, will generally not have the impact on a file that the CREATE overrides can have, so therefore those errors are tolerated.

30.02 IAM CREATE OVERRIDE STATEMENT FORMAT

The CREATE statement can be used to override file characteristics that are stored in a file and processing options that take effect when a file is defined or loaded. This override statement can be used either on the file definition or on the file load step. When used on a file load, the overridden file attributes will have precedence over attributes from the file definition.

Keywords that are related to file load, including BACKUPCOMPRESSED, CRBUFOPT, DATASPACE and TRACEDDNAME are only relevant to an actual file load. Conversely, the UNIT keyword is only relevant during a DEFINE. Note that the overrides will not be referenced for files being defined through JCL allocation. If overrides are desired for such files, they must be specified on the job step that is actually loading the file.

The following CREATE Override keywords are no longer documented in this manual. Information can be found on these keywords in the ICL library member OLDOVRID, which was downloaded as part of the installation procedure. The keywords are: BUFNO, DESCRIPTCODE and ROUTECODE.

The following CREATE Override keywords are no longer supported for the IAM VSAM interface. While a specification of any of these keywords will not produce an error message, the values provided will be ignored. Information can be found on these keywords in the ICL library member OLDOVRID, which was downloaded as part of the installation procedure. The keywords are: INFO, MINCOMPRESS, and SMF.

CREATE STATEMENT OPERANDS

CREATE	**DDNAME=**ddname **| DSN=**dataset name
[,BACKUPCOMPRESSED]	**[,BLKSIZE=**nnnnn**]**
[,COMPATIBLE]	**[,CORELIMIT=**nnnnn**]**
[,CRBUFOPT= <u>**MCYL**</u> **| CYL | MTRK | TRK | BSAM]**	
[,DATACOMPRESS= YES | NO | HW]	
[,DATASPACE=nnnn**]**	**[,DICTIONARY=**cccc**]**
[,ENHANCED]	**[,FIXED]**
[,INDEXCOMPRESS= YES | NO]	
[,INTEGRATED=nn**]**	**[,JRNAD= BOTH | BEFORE | AFTER | NONE]**
[,KEYLEN=nnn**]**	**[,LIMITKEYS=**nn**]**
[,LRECL=nnnnn**]**	**[,LOG= YES | NO]**
[,MAXBUFNO=nnnn**]**	**[,MAXSECONDARY=**nn**]**
[,MINBUFNO=nnnn**]**	**[,MULTIVOLUME= PRIMARY | SECONDARY]**
[,OCOREO%=xx**]**	**[,OCOREX%=**xx**]**
[,OVERFLOW=nnnnnnn**]**	**[,PE=**nnnnn**]**
[,PSEUDOLRECL=nnnnn**]**	**[,PSEUDORBA]**
[RELEASE= YES | NO]	**[,REREAD]**
[,RKP=nnnn**]**	**[,TRACEDDNAME=**ddname**]**
[,UNIT=unitname**]**	**[,VARIABLE]**
[,VAROVERFLOW= <u>**YES**</u> **| NO]**	**[WKDDNAME=**ddname**]**
[,XESDS]	

Figure 1: CREATE Override Statement

30.02 CONTINUED . . .

DDNAME= Specifies the DD name of the IAM data set that the override is to be applied to. Up to 40 DD names may be specified. If multiple DD names are specified they must be enclosed in parenthesis.

DD=&ALLDD will result in this override being applied as a global override to all IAM files being defined or loaded in this step, unless otherwise explicitly overridden.

When used during an IDCAMS DEFINE, it is recommended that DD=&ALLDD or the DSN= keyword be used. If using the DD= override on the define, the value MUST match the value specified for the FILE(ddname) keyword on the IDCAMS DEFINE control statement. In this circumstance, there is no need for the actual DD statement as specified by FILE(ddname) or DDNAME= on the CREATE override to be present in the JCL.

There is no default. Either DDNAME or DSN must be specified.

DSN= Specifies the data set name of the IAM data set that the override is to be applied to. One data set name per override control statement is permitted.

There is no default. Either DDNAME or DSN must be specified.

BACKUPCOMPRESSED Specifies that the input data is already in an IAM Data Compressed format. This can only be specified on a file load step with data that was created with the IAM ACCESS Override of BACKUPCOMPRESSED. Specification of this option forces the output file to have a data compressed format.

Default is that data is treated as standard, uncompressed data.

BLKSIZE= Specifies a blocking factor or an explicit block size for the IAM file. For values 1 to 15, IAM will use a block size that will result in the specified number of blocks per track for the device type on which the data set is being allocated to. For values of 300 or larger, IAM will use that value as an explicit block size. Values between 15 and 300 are invalid.

Default is generally 4 blocks per track, based on the IAM Global Option VSAMBLOCKF, the default blocking factor for IAM files defined under the VSAM interface. A larger block size may be used however if the DEFINE indicates a larger value for a Control Interval Size, or fewer than 5 records will fit in the default block size.

COMPATIBLE Specifies that the IAM file being defined or loaded is to have a format that is compatible with versions 6.2 of IAM or earlier. This will result in the file having a preformatted and allocated Independent Overflow area, and Prime Extension area. The opposite keyword value is ENHANCED, as described below.

Default is based on the IAM Global Options Table, which is shipped as Enhanced.

CORELIMIT= Specifies the minimum size that the index must be for IAM to use index compression. Using index compression can significantly reduce the amount of virtual storage IAM requires to process a data set, however if there is a substantial amount of random I/O, the CPU time may be higher with the compressed index.

Default value is from the IAM Global Options Table, which is shipped with default of 8000 bytes.

30.02 CONTINUED . . .

CRBUFOPT= Specifies the buffering option to be used for the file load. Because the file load is essentially a sequential output process, the buffering is different than the normal IAM Real Time Tuning Buffering. The MINBUFNO and MAXBUFNO values are not used during the file load. During the file load, IAM acquires a number of buffers based on this parameter. Once half the buffers are filled with data, then they are written out to the device in a single I/O operation. When the second group of buffers are filled, IAM will wait for the prior I/O to complete, then start the I/O for the second group, and let the application begin filling up the first group of buffers. The buffers are requested from extended private storage. This keyword value is only utilized for a file load step, it is ignored for a file define. The values that can be specified are:

BSAM – Use the BSAM Access Method macros to load the file. This value can only be used for Compatible format files. The default number of buffers acquired are 12, unless overridden by the BUFNO override. These buffers are all acquired in below the line storage.

CYL – Buffers for one cylinder's worth of blocks is acquired, approximately 1/2 cylinder is written per I/O.

MCYL – Buffers for two cylinder's worth of blocks is acquired, one cylinder is written per I/O.

MTRK – Buffers for ten tracks are acquired, five are written per I/O.

TRK – Buffers for two tracks are acquired, one track is written per I/O.

Default value is MCYL, or as otherwise specified in the IAM Global Options Table.

DATACOMPRESS= Specifies whether or not IAM is to attempt to compress the data content of each record. **Valid values are YES, NO, or HW.**

Specifying **YES** will use IAM software data compression. This will usually result in significant savings of DASD space for the file, reduce virtual storage requirements to process the file, and reduce physical I/O (EXCP's) for the file. Files must have at least 10 bytes of data after the key to be eligible for data compression. If the IAM data compression does not reduce the size of a record, it is placed in the file uncompressed.

Specifying **HW** will use the hardware compression instruction. This will require a dictionary. IAM will provide a default dictionary, or the user can create a compression dictionary and specify the name using the DICTIONARY override. Customers may be able to achieve greater compression through the use of the hardware compression facility, particularly if they create a compression dictionary from the data within the file. Hardware compression should not be used on processors prior to the IBM z/Series due to the significant CPU overhead on the older processors. The hardware compression overhead has been substantially reduced on the IBM z/Series processors, and may in some circumstances use less CPU time than the IAM software compression.

Default is files that are defined with a primary space of 75 tracks (5 cylinders) or larger will automatically be eligible for IAM software data compression. This file size can be changed in the IAM Global Options Table.

DATASPACE= Specifies the size, in megabytes, of the Data Space to be used for the temporary storage of the index to the IAM file which is being loaded. This override is ignored during file definition. Valid values are from 0 to 2048. A value of 0 results in the use of a dynamically allocated temporary data set, or other dataset that may be specified via the WKDDNAME override and DD card.

Default value is 256 megabytes, or as otherwise specified in the IAM Global Options Table.

30.02 CONTINUED . . .

DICTIONARY= Specifies the four-character suffix for the name of the user provided hardware compression dictionary. The dictionary must be in load module format, with the first four characters being 'IAMD', and the second four characters will be the customer's choice. Review the section on Hardware Compression for information on creating and naming the compression dictionaries.

Default is to use the IAM provided default dictionary.

ENHANCED Specifies that IAM is to create this file in the Enhanced format. This format eliminates the preallocated and preformatted Independent Overflow area, and Prime Extension area. Enhanced format KSDS files offer various processing enhancements including the ability to dynamically acquire additional DASD space for updates and inserts that occur after the file is loaded. The use of Enhanced format files is required for IAM ESDS files, for IAM files that have an alternate index, or for IAM files that will be used with the IAM RLS facility.

Default is Enhanced format file, unless otherwise specified in the IAM Global Options Table.

FIXED Applicable only to Compatible format files, specifies that the file has fixed length records. It may be necessary to provide this keyword, for Compatible mode files that are processed with the IAM Native or IAM ISAM interfaces when the Global Options of Enhanced file format and Data Compression will force files to have variable length records.

Default value is dependent on a variety of Global Options settings and define statement parameters, however it usually ends up defaulting to variable length records.

INDEXCOMPRESS= Specifies whether or not the file is eligible for a compressed index. Valid values are YES or NO.

When YES is specified, the file still has to meet the minimum size requirement as indicated by Global Options or CORELIMIT override, the key length must be between 4 and 128, and the index compression must yield storage savings of at least 10%.

When NO is specified, a compressed index will not be created for the file. For files with high random I/O activity, or with a very large number of records being randomly inserted into the file, not using the compressed index may reduce CPU time, at the expense of potentially substantial increased virtual storage requirements.

Default value is YES.

INTEGRATED= Specifies the amount of space, as a percentage, that is to be reserved for future growth within each prime data block during file load. This field is identical to the VSAM CI Freespace parameter. This value is also used for prime extension blocks as records are being added to the logical end of the file. Valid values are from 0 to 99.

Default, for KSDS files, is from the CI FREESPACE value specified during the file define. For ESDS files, the default is 0, or as otherwise specified in the IAM Global Options Table for ESDSINTEGRATED.

30.02 CONTINUED . . .

JRNAD= For Enhanced format files, indicates that the type of journalling, if any, that IAM will perform. When specified on the CREATE override, either during file definition or file load, the journalling specification will be used for any file update operation. Journalling can also be overridden by an IAM ACCESS override statement.

For files not being processed by IAM RLS, users must pre-allocate and catalog a log data set to be used for the journalling, which is required to be the name of the IAM data set / cluster, appended with the characters ".LOG". IAM journalling will not be active during file loads, reorganizations, or during recovery from the journal.

For files being processed by IAM RLS, the users must specify the journal data sets to the IAM RLS startup procedure.

One of the following values can be specified:

BOTH – The IAM log data set will contain both before and after images. This will enable the user to perform either a forward recovery, or a backward (backout) type of recovery.

BEFORE – The IAM log data set will contain only before images of updated records. This option allows backwards (backout) recoveries only.

AFTER – The IAM log data set will contain only the after images of updated records. This option allows forward recoveries only.

NONE – The IAM journalling feature will not be used for this IAM data set.

Default value is NONE.

KEYLEN= This override is provided for downward compatibility only. For file loads, this will override the key length that was specified when the file was defined.

Default value is the value specified when the file was defined.

LIMITKEYS= Specifies a value between 3 and 64, which will indicate the number of keys that will be in a set to be compressed. The total length of the keys within a set will be limited to the smaller of this value, or the number of keys that will fit within 256 bytes. Varying this number may yield better key compression, resulting in a smaller index, for some files. General recommendation is to use the default value.

Default value is 32.

LRECL= This override is provided for downward compatibility only. For file loads, this will override the maximum record length that was specified when the file was defined. If you provide this value as an override, you may need to increase the value by 4 if the file has variable length records, is data compressed, or is an Enhanced format file.

Default value is the value specified when the file was defined.

LOG= Indicates whether or not the IAM Override processor is to display this override card on the job log. Valid values are YES or NO.

Default value is NO, the override is not displayed unless it is in error.

MAXBUFNO= Specifies the maximum number of buffers IAM will acquire during file processing. IAM's Real Time Tuning dynamically adjusts the number of buffers used for a file as I/O demands change, up to the limit specified or default to for MAXBUFNO. While this has no effect during the file load, the value is saved within the file, and will be used as a default value for the file when opened for input or update processing. Valid values are from 1 to 2048. For Compatible format files, the maximum that will be used is 32.

Default is that no maximum number of buffers will be associated with the file.

30.02 CONTINUED . . .

MAXSECONDARY= For Enhanced format files, specifies a multiplication factor to be used by the IAM Dynamic Secondary Space Adjustment feature. Once a file has five or more extents on a volume, the secondary space value will be increased by the MAXSECONDARY factor, unless it exceeds the primary space value or the amount of space available on the volume. Valid values are from 0 to 10. A value of 0 will disable this feature.

Default value is 10 during file load, unless otherwise specified in the IAM Global Options table.

MINBUFNO= Specifies the minimum number of buffers that IAM will maintain during file processing. IAM's Real Time Tuning dynamically adjusts the number of buffers used for a file as I/O demands change, but will never use less than the specified or defaulted value for MINBUFNO. Valid values are from 1 to 2048. Note that for Compatible format files, the maximum is 32. Specifying an equivalent number of buffers for MAXBUFNO will effectively disable IAM's Real Time Tuning algorithm. If MINBUFNO is overridden, then that value will be used as the initial number of buffers to acquire during Open processing.

Default is that no minimum number of buffers will be associated with the file.

MULTIVOLUME= For **Enhanced** format files, indicates whether IAM is to use the PRIMARY or SECONDARY space allocation value, when IAM anticipates that the next extent will be obtained on another DASD volume. Valid values are PRIMARY or SECONDARY.

Default value is PRIMARY, based on the IAM Global Options Table.

OCOREO% = For **Compatible** format files, specifies the percentage of the total virtual storage required for the index to the Independent Overflow that will be acquired during file open for update. IAM acquires sufficient storage to contain the index for the number of records in Independent Overflow at Open plus this percentage of the total capacity. Valid values are from 0 to 100. When a large number of updates and inserts are anticipated, then this value should be increased to guarantee that the virtual storage will be available when needed. If virtual storage is not constrained, then smaller values may be of benefit for greater efficiencies in storage usage and processor cycles. If this keyword is specified on the CREATE override, then the value will become the default OCOREO% value for this file whenever it is opened for update.

Default is 10%, unless specified otherwise in the IAM Global Options Table.

OCOREX% = For **Compatible** format files, specifies the percentage of the total virtual storage required for the index to the Independent Overflow that will be acquired when additional storage is needed. Values from 1 to 100 may be specified. IAM will never acquire more than the total virtual storage required for the entire Independent Overflow area. If this keyword is specified on the CREATE override, then the value will be used as the default value for the file.

Default is 10%, unless otherwise overridden in the Global Options Table.

30.02 CONTINUED . . .

OVERFLOW= For **Compatible** format files, specifies the size of the Independent Overflow area in records. The Independent Overflow area is preformatted by IAM when the file is being loaded or reorganized. The Overflow area is restricted to 64,000 blocks . Because the override processor accepts the value in records, and does not know how many records will be placed within a block, it will accept a value of up to 3,000,000 records. If the specified value results in requiring more than 64,000 blocks, the file load will fail.

Default value is based on %CA Freespace and primary space allocation.

For **Enhanced** format files, this value will be used to calculate the target amount of DASD space to reserve for future expansion at the end of the file load. Unlike the function for Compatible format files, this space will not be formatted, and will not guarantee that the overflow area can actually hold this quantity of records. DASD space will only be reserved within the confines of the currently allocated space. No additional space will be acquired to meet this target. Additionally, the Overflow override will be used when providing information about available space for SHOWCB requests. When a file reaches the number of specified overflow records, the available space value will be zero, however records may still be able to be added to the file if there is DASD space available.

Default value is based on %CA Freespace and the amount of DASD space actually used for the file at completion of the load.

PE= For **Compatible** format files only, specifies the number of blocks that IAM is to reserve at the end of the file. This area acts as an extension to the prime area, which will handle inserts with keys higher than the existing prime records. Valid values are from 0 to 32,767. IAM will always increase the value specified by 1, except for the case where a value of PE=0 is specified, and the file requires only one prime data block. In that case, no PE blocks are allocated so that the file will use only one track with a blocking factor of 4. Also, for a single record load, IAM will dynamically increase the size of the Prime Extension area, except when PE=0 is specified.

For **ALL** IAM file types, specification of PE=0 will circumvent normal VSAM rules by allowing a file to be loaded with 0 records. Standard VSAM requires at least one record to be inserted into a file before it is considered to be in the loaded state.

Default is 3, which results in 4 PE blocks.

PSEUDOLRECL= Specifies the record length which IAM will pass back on SHOWCAT and SHOWCB requests as the maximum record length for this file. This is useful for applications that have a very large theoretical maximum record size, but have no records that are of the maximum length. Specify the smaller, actual maximum record size on the DEFINE parameters. Then, specify the larger theoretical maximum record size with the PSEUDOLRECL override. This will allow programs with the larger record layout to successfully open and process the file, while helping to conserve DASD space. This override ***MUST*** be specified on the job step that defines the file to take effect, and can be used on either Compatible or Enhanced format files.

With the use of the Variable Length Overflow support and spanned record support, this override is not as useful as it had been previously, and is provided for compatibility purposes only.

Default is to return the defined maximum record length.

30.02 CONTINUED . . .

PSEUDORBA Specifies that an IAM ESDS file can exceed 4 gigabytes in size, through the use of non-standard RBA's (Relative Byte Addresses). If the application has dependencies on the RBA to be identical to VSAM, then this parameter must not be used. This is helpful for applications that exceed the 4-gigabyte data size, and would require changes to support an 8-byte RBA value.

Default is standard VSAM 4-byte RBA.

RELEASE= Specifies whether IAM is to release unused space after the file is loaded, or reorganized. By default, IAM will release unused space in a file's allocation after the initial load, providing that secondary space was specified. For Compatible format files, the space required for Independent Overflow and PE is incorporated into the used area so that space will NOT be released. For Enhanced format files, IAM will reserve some, if any is available, of the over allocated space based on CA% Freespace, or the Overflow override if it is specified. Valid values are:

YES – IAM will release unused, unreserved space after each load or reorganization.

NO – IAM will not release unused DASD space. This may be useful for Enhanced format files, to hold on to DASD space for future expansion.

Default is to release unused, unreserved DASD space on the first load only, and only if a secondary space quantity has been specified. The default can be changed in the IAM Global Options Table.

REREAD When specified, indicates that IAM will reread the overrides each time an IAM file is opened. If specified, this keyword should appear on all of the override cards supplied in the file. This is primarily intended to be used by long running jobs, such as CICS regions, where the IAMOVRID points to a sequential data set or PDS member. This provides a mechanism for users to change overrides, and have them take effect by closing and reopening the IAM file(s).

Default is that the IAM override parameters are read when the first IAM file in a job step is opened, and kept in a table in virtual storage.

RKP= This is provided for downward compatibility only. When specified for a file load, this value indicates that IAM will use the value provided as the relative key position for the indicated file.

Default is that IAM will use the key offset value specified when the IAM file was defined.

TRACEDDNAME= Activates the IAM request trace capability, and indicates where the trace data is to be written to. Normally, it is written to SYSOUT, however it can also be written to a disk or tape file. To generate a printed report, specify a DD name of IAMTRPRx, where 'x' is any valid character. If 'x' is a 'C', then the key of each record is printed in character format, otherwise the key is printed in hexadecimal format.

WARNING: The trace to a particular output trace data set can only be done from a single IAM file at a time. Please DO NOT SPECIFY TRACEDDNAME WITH DD=&ALLDD. This can result in abends if multiple concurrent traces for different IAM files to the same TRACEDD are being performed.

Default is no trace output is produced.

30.02 CONTINUED . . .

UNIT= For a file define request, specifies a generic unit name for non-specific volume allocation. Disk volume(s) associated with the specified generic name must be mounted as STORAGE, and the VOLUMES parameter must specify 'ANYVOL' to activate IAM's non-specific volume allocation. This feature cannot be used for DFSMS managed data sets, or for CA-Vantage Allocation Manager, formerly Sterling's SAMS (VAM) managed data sets.

Default is SYSDA, unless otherwise specified by the WORKUNIT field in the IAM Global Options Table.

VARIABLE A keyword that specifies variable length record format for Compatible format files. Note that IAM Data Compressed files and Enhanced format files always have variable length records. Variable length records can also be set as the default in the IAM Global Options Table with the RECFM keyword.

VAROVERFLOW= For Enhanced format files, specifies whether or not IAM is to enable variable length record support for the overflow area. **Once a file has variable overflow enabled, it will not be disabled without doing a reorganization.** Valid values are YES or NO.

Default is YES, unless specified differently in the IAM Global Options Table.

WKDDNAME= Specifies the DD name of the DD statement specifying a temporary sequential data set used as temporary storage for the file's index as it is being loaded. This is generally not needed, as IAM normally uses either a data space, or it's own dynamically allocated temporary data set. However, for cases where the data space usage is not allowed, or cannot be large enough, and the default space allocation of the temporary work data set is insufficient, a user can provide their own data set allocation for this purpose.

Default if using a work data set is IAMWKDD.

XESDS For ESDS type of files, enables the use of 8 byte RBA values. This will allow ESDS files to exceed the 4 gigabyte file size with compatibility to DFSMS 1.5. If the Extended Addressability option has been set in the DFSMS Data Class for this ESDS file, then 8 byte RBA support is automatic. This option must not be specified if PSEUDORBA has been specified.

Default value is that IAM will generate a 4 byte RBA.

30.03 IAM ACCESS OVERRIDE STATEMENT FORMAT

The ACCESS statement is used to override IAM execution time defaults, for a specific step. The ACCESS override statement applies to file access processing only, it does not apply to file define, load, or create.

The following ACCESS Override keywords are no longer documented in this manual. Information can be found on these keywords in the ICL library member OLDOVRID, which was downloaded as part of the installation procedure. The keywords are: DESCRIPTCODE, LOG, and ROUTECODE.

The following ACCESS Override keywords are not supported for Enhanced Format IAM files. They can still be specified and used on the ACCESS override for COMPATIBLE format files. Information can be found on most of these keywords in the ICL library member OLDOVRID, which was downloaded as part of the installation procedure. The keywords are: BROWSE, BUFNO, COMPRESS, CORE, COREOUND, INDEXCOMPRESS, INFO, INTEGRATED, LSR, MODE, OPTCD, and SMF. Specification of any of these keywords will not cause an error, but the values will be ignored for Enhanced Format IAM files.

ACCESS STATEMENT OPERANDS

ACCESS **DDNAME=**ddname	**| DSN=**dataset name
[,BACKUPCOMPRESSED]	**[,DEFERWRITE= YES | NO]**
[,DJB = YES | NO]	**[,DYNCORE=**nnnn**]**
[,INDEXSPACE= YES | NO]	**[,JRNAD= BOTH | BEFORE | AFTER | NONE]**
[,LOG = YES | NO]	**[,MAXBUFNO=**nn**]**
[,MAXREGION=nnn**]**	**[,MAXSECONDARY=**nn**]**
[,MINBUFNO=nn**]**	**[,NORLS]**
[,OCOREO%=nn**]**	**[,OCOREX%=**nn**]**
[,REREAD]	**[,REREADEMPTY= YES | NO]**
[,RLS]	**[,SHAREOPTION = 1 | 2 | 3 | 4]**
[,TRACEDDNAME=ddname**]**	**[,TRACEREQUEST=** trace entry type(s) **]**
[,UPDATENQ= EXCL | SHR | NONE]	
[,VAROVERFLOW= YES | NO]	

Figure 2: Access Override Statement

30.03 CONTINUED . . .

DDNAME= Specifies the DD name of the IAM data set that the override is to be applied to. Up to 40 DD names may be specified. If multiple DD names can be specified, they must be enclosed in parenthesis.

DD=&ALLDD will result in this override being applied as a global override to all IAM files being accessed in this step, unless otherwise explicitly overridden.

There is no default. Either DDNAME or DSN must be specified.

DSN= For **Enhanced** format files only, specifies the data set name of the IAM data set that the override is to be applied to. One data set name per override control statement is permitted. Compatible format files are supported only by DDNAME on the ACCESS override.

There is no default. Either DDNAME or DSN must be specified.

BACKUPCOMPRESSED For Enhanced format files, specifies that IAM is not to decompress the data when passing it back to the requester. The data will be passed back to the requester in an IAM data compressed format. The data can be used for backup and reorganization purposes, but it is not usable by application programs. The data must be reloaded into an IAM file with the BACKUPCOMPRESSED IAM CREATE override specified, or the data must be decompressed with the IAMRECVR utility.

Default is that the data is returned in normal, uncompressed format.

DEFERWRITE= Specifies whether or not IAM will immediately write out buffers. Valid values are YES or NO. Specifying this will override the default for deferred writing, which is based on the share options the file was defined with, and whether or not the file is opened under CICS. Specifying YES will indicate that IAM will delay writing out an updated block until the buffer it resides in is required for another block, or until file close. Specifying NO will indicate that IAM will always immediately write out any updated data blocks.

Default is YES when a file has Share Option of 1 or 2, and is being accessed by a batch job. The default is NO when a file has Share Option of 3 or 4. For CICS, all randomly updated, inserted, or deleted records will cause an immediate write.

DJB= Specifies whether or not IAMRLS will perform Dynamic Job Backout processing for this job. Valid values are YES or NO. Specifying YES will indicate that IAM will backout any records changed by this job step for recoverable IAM files that are processed by IAMRLS should this job step abend. Specifying NO will indicate that the backout will not automatically occur, the user will have to use IAMJREST or other recovery process.

Default is the IAMRLS Dynamic Job Backout action specified on the IAMRLS startup parameters.

DYNCORE= Specifies an amount of memory for IAM's Dynamic Table. The value is specified in 1K (1024 byte) increments. IAM will attempt to GETMAIN the requested quantity during open, and if successful, will use that storage as a cache for randomly read records. On all random requests, where the complete key is specified, IAM will search the table for the record. These requests are identified on the IAMINFO report by R.(READ) requests for Compatible format files, or by GET RANDOM requests for Enhanced format files. If found, the record is returned without any physical I/O. If the record is not currently in the table, it will be read from disk, and placed in the table after it is retrieved.

Updates are always made both to the table and to the file. Variable length records are maintained as maximum length entries.

Valid values are from 0 to 16000 (i.e. Dynamic Table up to 16,000K bytes). The storage is requested from extended private.

Default value is 0, the Dynamic Table is disabled.

30.03 CONTINUED . . .

INDEXSPACE= For Enhanced format files, specifies whether or not IAM is to use an MVS Data Space to contain the prime and overflow index for this file. Use of this capability will, for most IAM files, substantially reduce the amount of extended private region used by IAM. The size of the Data Space is taken from the IAM Global Options Table DATASPACE value. Only one Index Space will be obtained per job step, however multiple IAM files are able to utilize the Index Space. Valid values are YES or NO.

Default is YES for CICS and NO for batch jobs, unless the default IAM Global Option has been changed.

JRNAD= For Enhanced format files, specifies the IAM journalling capabilities to be used. When specified on the ACCESS override statement, if the value specified enables journalling, the value will be combined with any specification from the CREATE override when the file was defined or loaded.

For files not being processed by IAM RLS, users must pre-allocate and catalog a log data set to be used for the journalling, which is required to be the name of the IAM data set / cluster, appended with the characters ".LOG". IAM journalling will not be active during file loads, reorganizations, or during recovery from the journal.

For files being processed by IAM RLS, the users must specify the journal data sets to the IAM RLS startup procedure if they want to use journalling.

The ACCESS override only changes the journalling value for the job step on which it is specified. Valid values are:

BOTH – The IAM log data set will contain both before and after images. This will enable the user to perform either a forward recovery, or a backward (backout) type of recovery.

BEFORE – The IAM log data set will contain before images of updated records. This option allows backward (backout) recoveries only. If AFTER images were specified on a CREATE override, then AFTER images will still be logged to the journal.

AFTER – The IAM log data set will contain after images of updated records. This option allows forward recoveries only. If BEFORE images were specified on a CREATE override they will still be logged to the journal.

NONE – The IAM journalling feature will not be used for this IAM data set. This will turn off any journalling for this job step, even if journalling had been specified on a CREATE override.

Default value is NONE, unless otherwise specified when the file was defined or loaded.

LOG= Specifies whether or not IAM will display this override card on the job log and in the job messages for this job. Valid values are YES or NO.

Default is that the override card is not displayed in the job messages, unless there is a syntax error on the override card.

MAXBUFNO= Specifies the maximum number of buffers IAM will acquire during file processing. IAM's Real Time Tuning dynamically adjusts the number of buffers used for a file as I/O demands change, up to the limit specified or default to for MAXBUFNO. Valid values are from 1 to 2048. Note that for Compatible format files, the maximum that will be used is 32.

Default is based on the larger of MAXBUFNO or BUFSP, or for CICS the larger of MAXBUFNO or CICISBUFSP from the IAM Global Options Table. As shipped, the default is the number of buffers that will fit within 875K of virtual storage, except for CICS where the default is 256K. For an Enhanced mode IAM file on a 3390, with quarter track blocking, the default MAXBUFNO is 65, except under CICS where it is 19.

30.03 CONTINUED . . .

MAXREGION= For Enhanced format files, specifies the maximum value, in megabytes, to which IAM Dynamic Region Adjustment will set the extended private region. This feature permits processing of files with large virtual storage requirements without the need to modify an installation's IEFUSI exit. Valid values are from 0 to 1024. A value of 0 will disable the Dynamic Region Adjustment feature.

Default value is 512 megabytes, or as otherwise specified in the IAM Global Options Table.

MAXSECONDARY= For Enhanced format files, specifies a multiplication factor to be used by the IAM Dynamic Secondary Space Adjustment feature. Once a file has five or more extents on a volume, the secondary space value will be increased by the MAXSECONDARY factor, providing that it does not exceed the primary space value, or the amount of space available on the volume. Valid values are from 0 to 10. A value of 0 will disable this feature.

Default value is 5 during file access, or as otherwise specified in the IAM Global Options Table.

MINBUFNO= Specifies the minimum number of buffers that IAM will maintain during file processing. IAM's Real Time Tuning dynamically adjusts the number of buffers used for a file as I/O demands change, but will never use less than the specified or defaulted value for MINBUFNO. Valid values are from 1 to 2048. Note that for Compatible format files, the maximum is 32. Specifying an equivalent number of buffers for MAXBUFNO will effectively disable IAM's Real Time Tuning algorithm. If MINBUFNO is overridden, then that value will be used as the initial number of buffers to acquire during Open processing.

Default value is 1.

NORLS Specifies that IAM is not to use the IAMRLS services for the identified data set(s).

Default action is based on IAM Global Options, and whether or not the IAMRLS address space is active.

OCOREO% = For **Compatible** format files, specifies the percentage of the total virtual storage required for the index to the Independent Overflow that will be acquired during file open for update. IAM acquires sufficient storage to contain the index for the number of records in Independent Overflow at Open plus this percentage of the total capacity. Valid values are from 0 to 100. When a large number of updates and inserts are anticipated, then this value should be increased to guarantee that the virtual storage will be available when needed. If virtual storage is not constrained, then smaller values may be of benefit for greater efficiencies in storage usage and processor cycles.

Default is 10%, unless specified otherwise in the IAM Global Options Table or overridden by an IAM CREATE Override.

OCOREX% = For **Compatible** format files, specifies the percentage of the total virtual storage required for the index to the Independent Overflow that will be acquired when additional storage is needed. Values from 1 to 100 may be specified. IAM will never acquire more than the total virtual storage required for the entire Independent Overflow area.

Default is 10%, unless specified otherwise in the IAM Global Options Table or overridden by an IAM CREATE Override.

REREAD When specified, indicates that IAM will reread the overrides each time an IAM file is opened. If specified, this keyword should appear on all of the ACCESS override cards supplied in the file. This is primarily intended to be used by long running jobs, such as CICS regions, where the IAMOVRID points to a sequential data set or PDS member. This provides a mechanism for users to change overrides, and have them take effect by closing and reopening the IAM file(s).

Default is that the IAM override parameters are read when the first IAM file in a jobstep is opened, and kept in a table in virtual storage.

30.03 CONTINUED . . .

REREADEMPTY Indicates whether or not IAM is to reread empty prime blocks when processing the file. When IAM is processing a file with REREADEMPTY=NO being specified or defaulted to, IAM will track the blocks that are detected as containing no data. Subsequent requests for those blocks will assume that they are empty, and therefore not do any I/O to read them. With REREADEMPTY=YES, IAM does not keep track of empty blocks, and will always read into storage the requested block.

Default action is based on the Cross Region Share Option, and the type of OPEN. For files defined with Share Option 1 opened for INPUT or UPDATE, and for files defined with Share Option 2 opened for UPDATE, the default is REREADEMPTY=NO. For all other cases, REREADEMPTY defaults to YES.

RLS Specifies that IAM is to process this file using the IAM RLS services. If the IAM RLS services are not available, then the OPEN will be failed.

Default is based on the IAM Global Options and whether or not the IAMRLS address space is active.

SHAREOPTION= For Enhanced format files, specifies the VSAM Cross Region Share Option to be used for this execution. This provides a mechanism for accessing the file with a different Share Option than it had been defined with. Share Options affects the Enqueues that are done to protect the file from concurrent access, and will also impact the buffering for the file. Refer to the IDCAMS section for a description of the various Share Options. Valid values are 1, 2, 3, or 4.

NOTE: SHARING IAM FILES FOR UPDATE IS STRONGLY DISCOURAGED, WITHOUT THE USE OF IAM RLS OR OTHER SOFTWARE THAT FACILITATES SHARING! SUCH SHARING MAY ADVERSELY AFFECT THE DATA INTEGRITY OF THE FILE WITHOUT THE PROPER SOFTWARE PROTECTION!

Default is that the file will be opened based on the Share Option specified when the file was defined.

TRACEDDNAME= Activates the IAM request trace capability, and indicates where the trace data is to be written to. Normally, it is written to SYSOUT, however it can also be written to a disk or tape file. To generate a printed report, specify a DD name of IAMTRPRx, where 'x' is any valid character. If 'x' is a 'C', then the key of each record is printed in character format, otherwise the key is printed in hexadecimal format.

WARNING: The trace to a particular output trace data set can only be done from a single IAM file at a time. Please DO NOT SPECIFY TRACEDDNAME WITH DD=&ALLDD. This can result in abends if multiple concurrent traces for different IAM files to the same TRACEDD are being performed.

Default is no trace output is produced.

30.03 CONTINUED . . .

TRACEREQUEST= Specifies the type of tracing to be done.

For **Enhanced** format files, allowable values are:

TRALL – Activates all of the trace points.

TRIOS – Trace start of each logical I/O request.

TRIOE – Trace at end of each logical I/O request.

TRBFR – Trace internal IAM calls to the IAM Buffer Manager.

TREXCP – Trace physical I/O requests.

TRXTND – Trace file expansion (extend) calls.

For Enhanced format files, multiple values for TRACEREQUEST can be specified by enclosing the desired keywords within parenthesis.

Default value is TRIOS.

For **Compatible** format files, allowable values are:

TRACE – Write a single TRACE record for each logical I/O Request. If the TRACEDDNAME specifies a DD name other than IAMTRPRx, then the output is assumed to be on tape or disk, and must be printed with the IAM IAMONRPT utility.

MONITOR – Write multiple MONITOR records for each request. The TRACEDDNAME must specify a tape or disk data set. The data must be printed by the IAM utility IAMONRPT. Note that MONITOR mode tracing is only supported for asynchronous I/O requests, such as issued by online systems, including CICS.

Default value is TRACE.

UPDATENQ= Overrides the ENQ processing that IAM will perform for the specified file. Normally, IAM will base the ENQ on the defined cross region share option. Valid values are:

EXCL – Indicates that IAM is to perform an exclusive ENQ on the file. This will prevent any other OPEN for this IAM file, within the scope of the ENQ capabilities provided on the host system.

SHR – Indicates that IAM is to perform a shared ENQ on the file. If specified for all users of the file, it will permit multiple jobs to have update capability to the same IAM file.

NONE – Prevents IAM from issuing any ENQ for the file being opened. The file will not be protected from concurrent update or even from a concurrent attempt to reload the file, while this application has the file open. This value should be specified when the file is being used by a job that is utilizing MVS Checkpoint / Restart facilities.

The default ENQ processing is based on the cross region share options, and the type of OPEN being issued for the data set.

VAROVERFLOW= For Enhanced format files, specifies whether or not IAM is to enable variable length record support for the overflow area. This override will only take effect when the file is opened for update. **Once a file has variable overflow enabled, it cannot be turned off until the file is reorganized.** Valid values are YES or NO.

Default is the value set when the file is defined or loaded, which defaults to YES, unless changed in the IAM Global Options Table.

30.04 IAM OVERRIDE STATEMENT EXAMPLES

The examples in this section are intended to help demonstrate how to use the IAM Override capability. There are other examples of using IAM Overrides throughout the manual, as the various features and capabilities of IAM are presented.

EXAMPLE 1: DEFINE WITH OVERRIDE

The following example shows how an IAM Override statement is used during an IDCAMS DEFINE of an IAM file, to request the file's data records be compressed. The IAM file's DEFINE will use the attributes of an existing VSAM cluster.

```
//DEFINE      EXEC  PGM=IDCAMS
//SYSPRINT     DD   SYSOUT=*
//IAMOVRID     DD   *
   CREATE      DD=&ALLDD,DATACOMPRESS=YES
/*
//SYSIN        DD   *
   DEFINE     CLUSTER           -
   (NAME(CICS.MASTER.$IAM)      -
   MODEL(CICS.MASTER.CLUSTER)  -
   VOLUMES(MVS001) )
/*
```

Figure 3: Example of IAM Override on Define

EXAMPLE 2: MULTIPLE OVERRIDES ON DEFINE

The following example shows an IDCAMS DEFINE of multiple IAM files, each with a different IAM Override values. As seen, this can easily be done through the use of the DSN parameter on the IAM Create Override statement.

The data set name "iam.master" will be a data compressed file, with Automatic Space Release turned off, due to anticipated file expansion. The data sets block size will be based on the IAM default value.

The data set named "iam.test" will use 1/2 track blocking, as indicated by the B=2 override. It will use the default for data compression and space release, as indicated in the IAM Options Table.

```
//DEFINE      EXEC  PGM=IDCAMS
//SYSPRINT     DD   SYSOUT=*
//IAMOVRID     DD   *
     CREATE   DSN=iam.master,DATACOMPRESS=YES,RELEASE=NO
     CREATE   DSN=iam.test,B=2
/*
//SYSIN        DD   *
     DEFINE   CLUSTER           -
        (NAME(my.iam.master     -
        OWNER($IAM) )           -
        CYLINDERS(100 20)       -
        FREESPACE(10 5)         -
        KEYS(24 8)              -
        RECORDSIZE(250 480)     -
        VOLUMES(IAM001) )
     DEFINE   CLUSTER           -
        (NAME(my.iam.test)      -
        OWNER($IAM) )           -
        CYLINDERS(10 5)         -
        FREESPACE(10 5)         -
        KEYS(8 8)               -
        RECORDSIZE(100 1250)    -
        VOLUMES(IAM001) )
/*
```

Figure 4: Example of Multiple Overrides on IDCAMS Define of Multiple Data Sets

30.04 CONTINUED . . .

EXAMPLE 3: FILE LOAD OVERRIDE

The following example demonstrates the use of an IAM Override for a file load, which is being done through IDCAMS. In particular, the CRBUFOPT override is specified to provide maximum buffering on the IAM file.

```
//LOADIAM   EXEC    PGM=IDCAMS
//SYSPRINT   DD     SYSOUT=*
//IAMINFO    DD     SYSOUT=*
//IAMOVRID   DD     *
     CREATE         DD=IAMFILE,CRBUFOPT=MCYL
/*
//IAMFILE    DD     DISP=OLD,DSN=my.iam.file
//INPUTFIL   DD     DSN=my.sequential.file,DISP=OLD,DCB=BUFNO=30
//SYSIN      DD     *
     REPRO  INFILE(INPUTFIL) OUTFILE(IAMFILE) REUSE
/*
```

Figure 5: Example of Overriding a File Load

USING IAM OVERRIDES WITH CICS

The following example shows IAM Overrides for files being processed under CICS. The IAM override cards are contained within a partitioned data set (PDS). The PDS must be defined with fixed length records of 80 bytes. The data set can be blocked at any valid block size for that record size. Please note that IAM will request that the buffers that are acquired by MVS for reading the override data set are to reside in extended private. Please note that the IAMOVRID and IAMINFO files do NOT have to be defined to CICS.

In order to provide increased flexibility, all of the ACCESS override cards contain the REREAD keyword. This will cause IAM to reread the override control file each time an IAM file is open, rather than just the first time an IAM file is opened. This provides the capability to change the override value(s), and have the new values take effect just by closing and reopening the particular IAM file. While REREAD only needs to be specified on one of the cards, by specifying it on all of the cards, any of them can be removed and still have the REREAD be effective. If the override cards are ever processed without the REREAD option being specified, then the file will not be reread. The implication of not specifying REREAD for CICS is that CICS will have to be shut down and restarted to utilize new override values.

The following override values are being specified. The MAXBUFNO value for files other than those otherwise overridden, is being set to 16. Two of the files will have their maximum buffers increased to 64 buffers. Another file will use the Dynamic Table (DYNCORE) option to table records in memory. A size of 1024K of memory is specified for the table. Also note that an IAM CREATE override is provided specifying CRBUFOPT=TRK. This is recommended to minimize the amount of storage used just in case an IAM file is opened empty under CICS, because the file load may use significant virtual storage resources. The contents of the control card file, for the above mentioned overrides is as follows:

EXAMPLE 4: CICS OVERRIDES IN A PDS

```
DSN=my.iam.override(cics1):
ACCESS        DD=(FILE1,FILE2),MAXBUFNO=64,REREAD
ACCESS        DD=FILE3,DYNCORE=1024,REREAD
CREATE        DD=&ALLDD,CRBUFOPT=TRK
ACCESS        DD=&ALLDD,MAXBUFNO=16,REREAD
```

Figure 6: Example of IAM Override cards in a partitioned data set

30.04 CONTINUED . . .

Below is an example of the JCL that could be used by CICS to utilize the above IAM Overrides. CICS can be executed here as either a started task, or as a batch job. Note that for started tasks, the IAM Overrides cannot reside in an instream data set, i.e. (DD *) is not allowed. Also note that the IAMOVRID is specified with a DISP=SHR, which will allow the file to be updated while CICS is running. While the IAM files here are shown as being allocated at CICS startup, the files could also be dynamically allocated by CICS.

EXAMPLE 5: CICS JCL WITH OVERRIDES

```
//CICS1     EXEC     PGM=DFHSIP
//.....
//FILE1      DD      DSN=IAM.FILE1,DISP=SHR
//FILE2      DD      DSN=IAM.FILE2,DISP=SHR
//FILE3      DD      DSN=IAM.FILE3,DISP=SHR
//....
//IAMINFO    DD      SYSOUT=*
//IAMOVRID   DD      DISP=SHR,DSN=my.iam.override(cics1)
```

Figure 7: Example of a CICS PROC/JCL with IAM Overrides in a PDS

USING OVERRIDES WITH ALTERNATE INDEX

When using IAM data sets with alternate indexes and paths, the DSN= parameter must be used, instead of the DDN= parameter. This is because the actual IAM data sets that are used will be dynamically allocated with system generated DD names, so the DD names are not known in advance. The only meaningful override for a PATH is the trace information. If tracing is specified for a PATH, tracing will be propagated to all of the data sets opened because of the PATH open. Other overrides on the PATH will be ignored.

The example below shows how to specify an override for a base cluster and an alternate index, which are being accessed via a PATH. The override for the base cluster will specify a MAXBUFNO value, while the override for the alternate index is using the IAM Dynamic Tabling feature. Use of dynamic tabling on the alternate index will provide a way to keep the alternate index in virtual storage, if it is not too large, and if it is being randomly accessed.

EXAMPLE 6: OVERRIDES WHEN USING A PATH

```
//UPDATEAIX     EXEC   PGM=UPDATE
//MASTERPTH      DD    DSN=iam.master.path,DISP=OLD
//IAMOVRID       DD    *
       ACCESS    DSN=iam.master.cluster,MAXBUFNO=64
       ACCESS    DSN=iam.master.aix,DYNCORE=1024
/*
```

Figure 8: Example of Overrides with an IAM Alternate Index

40.01 IAMSMFVS - DATASET ANALYSIS PROGRAM OVERVIEW

OVERVIEW

IAMSMFVS is a special purpose SMF data analysis program, intended to provide useful information in a concise format about indexed dataset activity. The reports can be used to:

1. Identify VSAM Clusters that are candidates for conversion to IAM.
2. Compare the results once the files are converted to IAM.
3. As a source for tracking IAM dataset activity to monitor IAM dataset usage.
4. Determine when IAM files need to be reorganized.

IAMSMFVS provides 4 different reports. The first report is a Summary report, which includes a summarization of VSAM and IAM activity so it is easy to tell the DASD space and I/O totals for each of the indexed access methods. The second report is an EXCP report, which is broken down by access method. There is a VSAM EXCP Report, and an IAM EXCP Report. Each one has the top 100 datasets for I/O activity for that access method. There is an option, DETAIL, which will provide a line for each SMF record for each of the datasets included in the EXCP report, with a breakdown of activity for that job step. The third report is a Dataset Summary Report, which contains information about each dataset for which information was gathered by IAMSMFVS sorted by the dataset name. The last report is based on dataset size, here again split by access method. Each size report has the largest 100 datasets for the specified access method.

SMF DATA REQUIREMENTS

The IAMSMFVS program analyzes dataset usage statistics gathered primarily from the Systems Management Facility (SMF records), with supporting data from the system catalog. SMF data may be taken from either the active SMF datasets or sequential SMF history datasets. To operate, IAMSMFVS requires SMF type 30, subtype 4 records, or type 4 records. For VSAM dataset reports, the SMF type 64 records must be collected. For the reports to include IAM datasets, the IAM SMF recording must be enabled and an SMF record type selected in the IAM Global Options table. IAMSMFVS does require the matching Step Termination record for each dataset record; otherwise the dataset record will not be processed. If the required SMF record types are not present, or have been modified from the standard MVS/ESA or OS/390 format, then IAMSMFVS will not be able to produce any reports.

IAMSMFVS reads the provided SMF data, and builds a table in virtual storage containing the extracted and accumulated data for each dataset. An IAM dataset requires one entry in the table. A VSAM dataset requires multiple entries, with one for the cluster level, and then an additional entry for each component. A typical VSAM KSDS will require 3 entries. The size of this table is controlled by the MAXDSNS keyword, which defaults to 1500. Once the dataset table is filled, no more datasets will be added to the table, however information on the datasets already in the table will continue to be accumulated.

If a VSAM dataset is not cataloged when IAMSMFVS is running then IAMSMFVS will not be able to associate the component names with a cluster name. Those VSAM components will be reported on just by the component name. To eliminate these stray VSAM components from the report, specify the CATONLY keyword.

40.02 IAMSMFVS JCL REQUIREMENTS

The JCL statements required to execute IAMSMFVS are as follows:

EXEC STATEMENT Specifies the IAM VSAM SMF ANALYSIS program name - IAMSMFVS. A large region parameter may need to be provided, based on the number of datasets that are being tracked. IAMSMFVS presently uses only below the 16-megabyte line storage.

DD STATEMENTS The following table describes the required DD statements for running IAMSMFVS. The amount of DASD space required for the temporary work files of SORTIN, SORTOUT, and SORTWKxx will depend entirely on the amount of SMF data that is being processed. A month's history tape is going to require much more DASD space than just a daily snapshot.

DDNAME	Description
STEPLIB or JOBLIB	Specifies the IAM Load Module Library. This statement may be omitted if the IAM modules are in a link list, as is recommended.
SYSPRINT	Specifies where the IAMSMFVS reports are to be printed. Usually a SYSOUT dataset.
SYSMF	Identifies the file containing the input SMF data. This may be an active SMF dataset or any file containing off loaded SMF data.
SYSIN	Specifies the control card input dataset. Usually a DD * dataset.
SORTIN	Specifies a work file that contains extracted SMF data that is used by IAMSMFVS to create the reports. This data will be passed to the external sort.
SORTOUT	Specifies a work file, which will be returned from the external sort routine.
SORTWKnn SORTLIB	If there are any DD statements needed by your sort program (i.e. SORTLIB, SORTWKnn, etc.), they must be included. Refer to documentation for your sort.
SYSPUNCH	IAMSMFVS can optionally generate IAMSIMVS SELECT statements. Whenever IAMSMFVS finds a SYSPUNCH DD statement is present in the JCL it will use the cluster names provided in the Cluster Size Report and create IAMSIMVS SELECT command statements. This DD statement should specify a sequential dataset or a member in a PDS capable of accepting 80 character records.

```
//SYSPUNCH DD DSN=SIMVS.INPUT,DISP=(,CATLG),
//            UNIT=SYSDA,SPACE=(TRK,(15,15))
```

40.03 IAMSMFVS - REPORT COMMAND

REPORT COMMAND The REPORT command is used to initiate IAMSMFVS processing. There are various options that can be specified, as described below. The input can be an SMF history file (RECFM=VBS) or a system SMF dataset. It is recommended that initially no operands be specified. The default reports would then list the 100 datasets, by dataset organization, with the most EXCP activity, and the 100 largest datasets, again by dataset organization.

REPORT	
[CATONLY]	**[,JOBNAMES=**ccccc **]**
[,CHECKLENTH]	**[,MAXDSNS=**nnnn **]**
[,CURRENT]	**[,MAXJOBS=**nnnn **]**
[,DETAIL]	**[,MAXRECLENGTH=**nnnnn **]**
[,DFEFERRPRT=ccc **]**	**[,MAXREPORTS=**nnnn **]**
[,DSGROUPS=ccccc **]**	**[,PRTLEN‘GTH=**nnnnn **]**
[,DSNAMES=ccccc **]**	**[,RECSIZE(rrr)=**nnnnn **]**
[,DSORG=ccc **]**	**[,SORTCORE=**nnnnn **]**
[,ERRORPRINT]	**[,SORTMSG=**cc **]**
[,FROMDATE=yyyyddd **]**	**[,SORTPFX=**ccccc **]**
[,FROMDDNAME=ddname **]**	**[,TEMPDSNAMES]**
[,FROMTIME=hhmmss **]**	**[,TODATE=**yyyyddd **]**
[,GROUPNAMES=ccccc **]**	**[,TOTIME=**hhmmss **]**

Figure 1: Report Command Operands

OPERANDS The following operands may be specified on the REPORT command. The underscored portion indicates the minimum abbreviation for the keyword.

Operand	Description
CATONLY	Specifies that IAMSMFVS will only report on VSAM datasets that are cataloged. This will eliminate from the report stray component entries for VSAM datasets that are not currently cataloged.
CHECKLENGTH	Specifies that SMF data records are to be validated against a table of minimum record lengths. If the user has modified the minimum record length or is executing in a non-compatible system, the correct record lengths may be specified by the RECSIZE operand. By default, the length of an SMF record is not validated.
CURRENT	Specifies that only the most recent values for overflow use in IAM files will be reported on, rather than the maximum amount used. This may be useful for determining when an IAM file needs to be reorganized. By default, the maximum amount of overflow use is reported on.
DETAIL	The EXCP Reports are to include detailed statistics by job step. By default, only summary dataset activity is reported.

40.03 CONTINUED . . .

DFEFERRPRT= Indicates whether or not error messages from the VSAM ICF identification processor should be printed. Valid values are:

AC – All messages to the console

NO – Error messages are not to be printed.

YES – Error messages are to be printed.

The default is NO.

DSGROUPS= Specifies that only records having dataset names which begin with the given character string(s) will be processed. This operand specifies a partial dataset name from 1 to 44 characters in length. Up to 100 dataset groups may be specified for a single command if entered as follows:

DSGROUPS=(dsg1,dsg2,...,dsgn)

If neither the DSGROUPS nor DSNAMES operand is specified, dataset selection will be based upon the value specified for DSORG (or its default).

DSNAMES= Specifies that only records having a dataset name which match the dataset name(s) specified will be processed. This operand specifies a complete dataset name from 1 to 44 characters in length. Up to 100 dataset names may be specified for a single command if entered as follows:

DSNAMES=(c...c,...,c...c)

If neither the DSGROUPS nor DSNAMES operand is specified, dataset selection will be based upon the value specified for DSORG (or its default).

DSORG= Identifies the dataset organization that is to be processed. Valid values are:

AM – only VSAM clusters.

IAM – only IAM datasets.

The default is AM (VSAM) and IAM.

NOTE: To extract data for IAM datasets, SMF recording of IAM SMF records must be enabled, as described in the IAM Users Manual in section 91

ERRORPRINT Specifies that any SMF data record that causes an error during processing or fails length verification is to be printed.

The default is records in error are not printed.

FROMDATE= Specifies the lower date limit of the SMF records that are to be analyzed, in the form 'yyyyddd' or 'yyddd'. The long form must be used for dates from the year 2000 and above. The short form is prefixed with '19'.

The default is that there is no lower limit on the date for record selection.

FROMDDNAME= Specifies the DDNAME of the SMF file that is to be used as input to IAMSMF.

The default input DDNAME is SYSMF.

40.03 CONTINUED . . .

FROMTIME= Specifies the lower time limit of the SMF records that are to be analyzed, in the form of 'hhmmss'.

When used with FROMDATE, forms a combined starting point of date and time.

When used without FROMDATE, the FROMTIME applies to all days for which SMF records are being processed.

The default is there is no lower time limit on the record selection.

GROUPNAMES= Specifies that only those SMF records having a job name which begin with the specified character string(s) will be analyzed. This operand specifies a partial job name from 1 to 8 characters in length. Up to 50 job groups and/or names may be specified for a single command if entered as follows:

GROUPNAMES=(jobn1,jobn2,jobn3,....,jobnx)

The default, if neither the GROUPNAMES nor JOBNAMES operand is specified, is that the job name will not participate in SMF record selection.

JOBNAMES= Specifies those only records having a jobname which match the jobname(s) specified will be copied. This operand specifies a complete jobname from 1 to 8 characters in length. Up to 50 names may be specified in a single command if entered as follows:

JOBNAMES=(jobname1,jobname2,...,jobnamex)

The default, if neither the GROUPNAMES nor JOBNAMES operand is specified, is that the job name will not participate in SMF record selection.

MAXDSNS= Specifies the maximum number of unique datasets and/or clusters that will be tabled during this execution. The number can be any value from 20 to 32000, inclusive.

The default is 1500 dataset table entries.

NOTE: Each IAM dataset takes one (1) entry while each VSAM cluster takes three (3) or more entries, depending upon the number of components.

MAXJOBS= Specifies the maximum number of unique job names that will be tabled during this execution. The number can be any value from 20 to 32000, inclusive.

The default is 5000 job name table entries.

MAXRECLENGTH= Specifies the largest SMF record that the program will process. The number may be any value from 16384 to 65536, inclusive.

The default is 16384 bytes.

MAXREPORTS= Specifies the maximum number of dataset and/or cluster names to be included in the EXCP Activity Report(s) and the Size Report(s). The number can be any value from 20 to 32000, inclusive. All datasets are included in the dataset report.

The default is 100.

PRTLENGTH= Limits the number of bytes of data to be printed if ERRORPRINT is indicated. The number may be any value from 32 to 65536, inclusive.

The default is 32768 bytes.

RECSIZE(rrr)= Establishes the minimum length of the SMF record type 'rrr' as the value 'nnnn'.

The default minimum record lengths for system generated SMF records are documented in the IBM Systems Management Facilities manual.

40.03 CONTINUED . . .

SORTCORE= Specifies the amount of storage for the SORT to use. The number maybe any value from 10000 to 8000000 inclusive.

The default is 100000.

SORTMSG= Specifies the message option to be used by the SORT. Valid values are:

AC – All messages to the console

AP – All messages to the printer (SYSOUT)

CC – Critical messages to the console

CP – Critical messages to the printer

NO – No messages to be produced

PC – Critical messages to both console and printer

The default is CC.

SORTPFX= Specifies the DDNAME prefix to be used by the SORT. If the string specified is less than 4 characters, a dollar sign ($) fill character will be used.

The default is SORT.

TEMPDSNAMES Specifies that reports produced by the REPORT command will contain information on temporary as well as permanent datasets.

By default, temporary datasets are ignored.

TODATE= Specifies the upper date limit of the SMF records that are to be analyzed. Must be in the form 'yyyyddd' or 'yyddd'. The shorter form implies a prefix of 19.

The default is that there is no upper date limitation for record selection.

TOTIME= Specifies the upper time limit of the SMF records that are to be analyzed, in the format of 'hhmmss'.

When used with the TODATE keyword, this forms a combined ending point of the date and time specified. When used without the TODATE keyword, the 'TOTIME' is applied to each day for which SMF records are being processed.

The default is that there is no maximum time

40.04 IAMSMFVS USAGE EXAMPLES

Shown below are some examples of how to run IAMSMFVS, which include JCL and the control card input. Examples and descriptions of the reports produced are in the following sections.

EXAMPLE A: BASIC IAMSMFVS REPORT

The first example demonstrates how to run IAMSMFVS to obtain the basic reports, which will include the 100 busiest VSAM datasets, and the 100 largest VSAM datasets. IAM dataset reports will also be produced, providing that the IAM SMF recording has been activated.

```
//REPORT   EXEC  PGM=IAMSMFVS,REGION=1M
//STEPLIB  DD    DISP=SHR,DSN=iam load library
//SYSPRINT DD    SYSOUT=*
//SYSMF    DD    DISP=SHR,DSN=smf data
//SORTIN   DD    UNIT=SYSDA,SPACE=(CYL,(10,10))
//SORTOUT  DD    UNIT=SYSDA,SPACE=(CYL,(10,10))
//SORTWK01 DD    UNIT=SYSDA,SPACE=(CYL,(10,10))
//SORTWK02 DD    UNIT=SYSDA,SPACE=(CYL,(10,10))
//SORTWK03 DD    UNIT=SYSDA,SPACE=(CYL,(10,10))
//SYSIN    DD    *
      REPORT
/*
```

Figure 2: Example 1 of Using IAMSMFVS (EX4004A)

EXAMPLE B: REPORTING ON MORE DATASETS

This next example demonstrates obtaining the IAMSMFVS reports with providing a larger capacity both in the number of datasets tabled (MAXDSNS=3000), and in the number of datasets included in the EXCP and Size Reports, (MAXREPORTS=300). By specifying MAXDSNS=3000, that should allow for up to 1000 VSAM KSDS clusters. DSORG=AM is specified to only report on VSAM datasets. Also, because there are some large step termination records, SMF Type 30 Subtype 4, the maximum SMF record length is being increased.

```
//REPORT   EXEC  PGM=IAMSMFVS,REGION=4M
//STEPLIB  DD    DISP=SHR,DSN=iam load library
//SYSPRINT DD    SYSOUT=*
//SYSMF    DD    DISP=SHR,DSN=smf data
//SORTIN   DD    UNIT=SYSDA,SPACE=(CYL,(10,10))
//SORTOUT  DD    UNIT=SYSDA,SPACE=(CYL,(10,10))
//SORTWK01 DD    UNIT=SYSDA,SPACE=(CYL,(10,10))
//SORTWK02 DD    UNIT=SYSDA,SPACE=(CYL,(10,10))
//SORTWK03 DD    UNIT=SYSDA,SPACE=(CYL,(10,10))
//SYSIN    DD    *
      REPORT MAXDSNS=3000,MAXREPORTS=300,MAXRECLENGTH=32768,
             DSORG=AM
/*
```

Figure 3: Example of Reporting on More VSAM Datasets (EX4004B)

40.04 CONTINUED . . .

EXAMPLE C: USING A DATE RANGE

In the next example, an SMF history tape is being processed. The FROMDATE and TODATE keywords are specified, to limit the report to only one particular week's worth of data. The space parameters for the temporary work datasets have been increased due to the volume of data that is being processed.

```
//REPORT   EXEC  PGM=IAMSMFVS,REGION=4M
//STEPLIB   DD   DISP=SHR,DSN=iam load library
//SYSPRINT  DD   SYSOUT=*
//SYSMF     DD   DISP=SHR,DSN=smf history data
//SORTIN    DD   UNIT=SYSDA,SPACE=(CYL,(100,10))
//SORTOUT   DD   UNIT=SYSDA,SPACE=(CYL,(100,10))
//SORTWK01  DD   UNIT=SYSDA,SPACE=(CYL,(100,10))
//SORTWK02  DD   UNIT=SYSDA,SPACE=(CYL,(100,10))
//SORTWK03  DD   UNIT=SYSDA,SPACE=(CYL,(100,10))
//SYSIN     DD   *
      REPORT     FROMDATE=1997312,TODATE=1997319
/*
```

Figure 4: IAMSMFVS Example with FROMDATE and TODATE (EX4004C)

EXAMPLE D: REPORTING ON IAM DATASETS

The following example demonstrates how to generate a report to help determine which IAM datasets need to be reorganized. The DSORG=IAM keyword is specified, because only IAM datasets are going to be examined. The IAM SMF recording must have been enabled so that there are records to report on. Make sure that the SMF parameters will actually store the specified record number, and that they are also collected to your SMF history tapes. To make sure that the most recent information is being used, the CURRENT keyword is specified. A FROMDATE is also being specified, as the input is an SMF history tape. All SMF records from the date specified will be considered in the analysis. That date probably would be the last date the report was run. The report that will be most useful in determining if reorganization is necessary is the IAM Size report. MAXREPORTS is being specified, because there are more than 100 IAM datasets in the shop.

```
//REPORT   EXEC  PGM=IAMSMFVS,REGION=4M
//STEPLIB   DD   DISP=SHR,DSN=iam load library
//SYSPRINT  DD   SYSOUT=*
//SYSMF     DD   DISP=SHR,DSN=smf history data
//SORTIN    DD   UNIT=SYSDA,SPACE=(CYL,(50,10))
//SORTOUT   DD   UNIT=SYSDA,SPACE=(CYL,(50,10))
//SORTWK01  DD   UNIT=SYSDA,SPACE=(CYL,(50,10))
//SORTWK02  DD   UNIT=SYSDA,SPACE=(CYL,(50,10))
//SORTWK03  DD   UNIT=SYSDA,SPACE=(CYL,(50,10))
//SYSIN     DD   *
      REPORT     FROMDATE=2002218,DSORG=IAM,CURRENT,
                 MAXREPORTS=500
/*
```

Figure 5: Reporting on IAM Datasets (EX4004D)

40.04 CONTINUED . . .

EXAMPLE E: IAMSMFVS DETAIL REPORT

Another useful way to run IAMSMFVS is with the DETAIL option. With that option, you can find out which jobs, steps, and programs are using the VSAM or IAM datasets, and what the level of I/O activity is. One of the main reasons for doing this is in planning the conversion of datasets from VSAM to IAM. This will help identify the job(s) that are actively using the dataset, and can serve as a basis for comparison after the conversion.

The example below shows the JCL and control card required to obtain a detailed report. This example also demonstrates the use of the DSG= operand, which is used to limit the number of datasets being reported on to only those that are of most interest. Using such a selection criteria will be helpful in reducing the amount of output from the execution, as well as to make sure that the report includes all of the datasets that you need to analyze.

```
//DETAIL   EXEC  PGM=IAMSMFVS,REGION=4M
//STEPLIB  DD    DISP=SHR,DSN=iam load library
//SYSPRINT DD    SYSOUT=*
//SYSMF    DD    DISP=SHR,DSN=smf history data
//SORTIN   DD    UNIT=SYSDA,SPACE=(CYL,(50,10))
//SORTOUT  DD    UNIT=SYSDA,SPACE=(CYL,(50,10))
//SORTWK01 DD    UNIT=SYSDA,SPACE=(CYL,(50,10))
//SORTWK02 DD    UNIT=SYSDA,SPACE=(CYL,(50,10))
//SORTWK03 DD    UNIT=SYSDA,SPACE=(CYL,(50,10))
//SYSIN    DD    *
      REPORT     DSG=PROD.ORDERENT,DETAIL,MAXREPORTS=200
/*
```

Figure 6: IAMSMFVS Example for a DETAIL Report (EX4004E)

40.05 IAMSMFVS SUMMARY REPORT

IAMSMFVS supplies the user with a SUMMARY and three reports. Those reports are the EXCP Report, the Dataset Summary Report, and the Size Report. Examples of each of the reports, along with explanations of the fields are presented below.

SUMMARY REPORT On the very first page, the REPORT control card being processed is printed. That is followed by summary information for the run.

```
IAM400 SMF REPORT/DATA EXTRACT PROGRAM-IAMSMFVS VER 8.0/01P-INNOVATION DATA PROCESSING DATE-2002.218
IAM303   CARD IMAGE - *   REPORT                                         00120036*
IAM491   SMF REPORT       FUNCTION STARTED - 13.29.13
IAM601   SMF RECORDS -- READ.....17741  USED......2364  DROPPED.........0
         149 DATASETS TABLED REPRESENTING  14 JOBS -- DATED 2002.217 18:21 THRU  2002.218 13:22
         SPACE UTILIZATION SUMMARY -
           DEVICE TYPE.....3380     VSAM CYL/TRK......342/03  IAM CYL/TRK.....8907/13
           DEVICE TYPE.....3390     VSAM CYL/TRK........4/13  IAM CYL/TRK....18196/11
     TOTAL DISK EXCPS......888896   VSAM EXCPS..........7671  IAM EXCPS........711058
```

Figure 7: Sample IAMSMFVS Summary Report

IAM601 MESSAGE The IAM601 message presents a summary of the input SMF data. This includes the number of records read from the input SMF file, the number of records used, and the number of records dropped. Used records indicate the number of Step termination records (type 30 subtype 4 or type 4) and the number of dataset records (IAM records, VSAM type 64 records) that met the selection criteria for this execution. The number of records dropped indicates the number of records that exceeded the maximum record length.

The following line indicates the number of datasets (VSAM and/or IAM) that were included in the table for analysis and the number of unique job names that were found. (Note that this is not an actual count of number of jobs, only of the number of unique job names.) This is followed by the date and time period from the SMF records that was found on the input file.

SPACE UTILIZATION SUMMARY The next set of information provided is a space utilization summary, based on DASD device type and dataset organization. This section will only include data for the dataset organizations that IAMSMFVS was processing data for. So, for example if you had requested only VSAM datasets (DSORG=AM), then no values would be provided for IAM datasets. The amount of space figure is based upon the maximum space for each dataset, as determined from the information in the SMF records.

TOTAL DISK EXCP'S Then an EXCP summary is provided. The Total Disk EXCP's comes from an accumulation of the information from the step termination records for all non-temporary DASD datasets. The figures for each access method are the totals from the actual dataset SMF records.

40.06 IAMSMFVS EXCP REPORTS

After the summary report appear the EXCP reports. These reports are organized by access method, and contain the datasets with the most EXCP's. The maximum number of datasets in each access method report is based on the MAXREPORTS keyword of the REPORT command, which defaults to 100. The different access method reports are quite similar, so samples of VSAM and IAM EXCP Reports will be shown, followed by descriptions of the various fields contained in the report.

VSAM EXCP REPORT

```
 IAM400 SMF REPORT/DATA EXTRACT PROGRAM-IAMSMFVS VER 8.0/01P-INNOVATION DATA PROCESSING 2002.218
                                                VSAM EXCP REPORT
                                   USE TOTAL                                                 SPLITS ALLOC
DATA SET NAME                    COUNT EXCPS RECORDS READS INSERTS UPDATES DELETES CI  CA   TRKS
---- --- ----                    ----- ----- ------- ----- ------- ------- ------- --  --   ----
SYSPCICS.CICSIDP1.DFHGCD             7  1727
  SYSPCICS.CICSIDP1.DFHGCD.DATA      7  1552     655  5116    1968     394    1963 27   0     15
  SYSPCICS.CICSIDP1.DFHGCD.INDEX     7   175       1     0       0      66       0  0   0      1
SYSPSMF.CPUB.MAN1                    5   900
  SYSPSMF.CPUB.MAN1.DATA             5   900       0 72546       0       0       0  0   0    900
SYSPCICS.CICSIDP1.DFHLCD             7   580
  SYSPCICS.CICSIDP1.DFHLCD.DATA      7   566     260  2008       0      81       0  0   0      4
  SYSPCICS.CICSIDP1.DFHLCD.INDEX     7    14       1     0       0       0       0  0   0      1
AJM.UPSTREAM.CATALOG                 2   373
  AJM.UPSTREAM.CATALOG.DATA          2   277    7017 13517      71       0      27  0   0     90
  AJM.UPSTREAM.CATALOG.INDEX         2    96       7     0       0       0       0  0   0      1
SYSPRMM.MASTER                       2   360
  SYSPRMM.MASTER.DATA                2   344    4015  6092       0       0       0  0   0    285
  SYSPRMM.MASTER.INDEX               2    16       9     0       0       0       0  0   0      1
IAM63.VSAMTEST.DATA                 81   243
  IAM63.VSAMTEST.DATA.DATA          81   162    1172    81       0      81       0  0   0     45
  IAM63.VSAMTEST.DATA.INDEX         81    81       1     0       0       0       0  0   0      1
```

Figure 8: Sample VSAM EXCP Report

IAM EXCP REPORT

```
 IAM400 SMF REPORT/DATA EXTRACT PROGRAM-IAMSMFVS VER 8.0/01P-INNOVATION DATA PROCESSING 2002.218
                                          IAM EXCP REPORT
                             USE  TOTAL                                                 OVRFLW    USED
DATA SET NAME              COUNT  EXCPS  RECORDS    READS INSERTS UPDATES DELETES  RECS   %   TRKS
---- --- ----              -----  -----  -------    ----- ------- ------- -------  ----  ---  ----
IAMV.VIT551.CLUSTER            2 203352  2000000   528000       0       0       0     0 100 50086
IAMV.VIT552.CLUSTER            2 135086  2000000  2462001       0       0       0     0 100 50366
IAMV.P630232.$IAM.CLUSTER      2  65429  5040000   280000   20000   40000       0 15761   1  8803
IAMV.P630176.CLUSTER           2  62520    64801    21000   32111       0       0 22750  95   813
IAMV.P630119.CLUSTER           2  13704  1716000  1716001       0 1716000       0     0 100 33016
IAMV.VXA517.CLUSTER            2  13613    26892    41310    1385    1237     266   725  14   162
IAMV.VIT194.CLUSTER            6  12796    23000    52002    8000       0       0  4000  80   107
IAMV.P630060.CLUSTER           3  12608    10000    39993    4000       0    8000     0 100   251
IAMV.VIT548.CLUSTER            2  11747    27150    41311    1385    1237     137   725  14   162
IAMV.P630131.CLUSTER           3  10922   215000   115001  100000       0       0     0   0  2887
IAMV.P630231A.CLUSTER          2   8479    50000     4599       0    4599       0     0 100  1252
IAMV.P630231.CLUSTER           2   8451    50000     4599       0    4599       0     0 100  1252
IAMV.P630076.CLUSTER           4   7846    50001    25002   17007       0       0  7998  21  1693
IAMV.VIT547.CLUSTER2           1   7393    50000    16000       0       0       0     0 100  1253
IAMV.VIT194CP.CLUSTER          3   6398    23000    26001    4000       0       0  4000  62   107
IAMV.P630075.CLUSTER           3   5521    50001        0   17007       0       0  7998  21  1693
IAMV.P630187.CLUSTER           3   4788    72000      500    6396       0       0 24000  16  1020
IAMV.P630162A.CLUSTER          3   4788    72000      500    6396       0       0 24000  16  1020
```

Figure 9: Sample IAM EXCP Report

40.06 CONTINUED . . .

EXCP REPORT FIELD DESCRIPTIONS

A description of the fields appearing on the EXCP reports is provided below. Note that some fields represent an accumulated value, whereas other fields indicate a maximum value. For IAM, maximum values for overflow are replaced with the most recent values if the keyword CURRENT is specified on the REPORT command.

Field Name	**Description**
DataSet Name	Specifies the name of the dataset. For cataloged VSAM datasets, this will have the cluster name on the first line, followed by the component names on subsequent lines, which are indented. For uncataloged VSAM datasets, only the component name will be printed.
Use Count	Indicates the number of SMF records processed for this dataset. This will usually be the number of times the dataset was closed during the time interval of the report.
Total EXCPS	The accumulated total EXCP counts from all of the SMF records processed for this dataset. For cataloged VSAM clusters, this number will be the total for all of the components, with the component value given on the subsequent lines.
Records	Contains the maximum number of records that were in the file.
Reads	Accumulated total from all of the SMF records found for the specified dataset, of the records read.
Inserts	Accumulated total number of records inserted. This value does NOT include the count of records from an initial file load. Also, for VSAM clusters, does not include records added to the end of the file.
Updates	Accumulated total number of records updated.
Deletes	Accumulated total number of records deleted.
Splits – CI	For VSAM components, the accumulated total number of CI splits that occurred.
Splits – CA	For VSAM components, the accumulated total number of CA splits that occurred.
Alloc Tracks	For VSAM components, the maximum number of tracks allocated to the component.
Overflow Recs	For IAM datasets, indicates the maximum number of records in overflow, unless the keyword CURRENT was specified on the REPORT command. In that case, it will contain the value from the most recent SMF record for the dataset.
Overflow %	For IAM datasets, indicates the maximum percentage, or most recent value if CURRENT was specified, of the overflow area that was used. If a file has no overflow area, then this number will be 100 even though there are no records in the overflow area. For Enhanced format files that are defined with an Overflow override value, the percentage is based on that number. For Enhanced format files without an Overflow override, value is based on assumption using a total value for overflow from overflow blocks currently used plus all of the unused extended blocks. For Compatible format files, this is the percentage is based on the established size of the Independent Overflow Area.
Used Tracks	For IAM datasets, indicates the maximum amount of DASD space used for this file.

40.06 CONTINUED . . .

DETAIL REPORT A more detailed EXCP Report on each dataset's activity is available by specifying the DETAIL keyword on the REPORT control card. This report will include a line for each SMF record encountered for each dataset included on the EXCP report, with the activity statistics plus job name, step name, and program name.

```
IAM400   SMF REPORT/DATA EXTRACT PROGRAM - IAMSMFVS VER 8.0/00T  - INNOVATION DATA PROCESSING DATE - 2002.218 PAGE
                                                        IAM EXCP REPORT
                                                 USE     TOTAL                                                OVRFLW    USED
DATA SET NAME                                   COUNT    EXCPS  RECORDS   READS INSERTS UPDATES DELETES  RECS   %   TRKS
---- --- ----                                   -----    -----  -------   ----- ------- ------- -------  ----  ---  ----
IAMV.VIT551.CLUSTER                                 2   203352  2000000  528000       0       0       0     0  100 50086
2002.217 J=JFMTESTS S=VIT551A  P=SUBTMNGR D=VSAMCRT1    3341  2000000       0       0       0       0     0  100 50086
         J=JFMTESTS S=VIT551A  P=SUBTMNGR D=VSAMCRT9  200011  2000000  528000       0       0       0     0  100 50086

IAMV.VIT552.CLUSTER                                 2   135086  2000000 2462001       0       0       0     0  100 50366
2002.217 J=JFMTESTS S=VIT552A  P=SUBTMNGR D=VSAMCRT1    3359  2000000       0       0       0       0     0  100 50366
         J=JFMTESTS S=VIT552A  P=SUBTMNGR D=VSAMCRT9  131727  2000000 2462001       0       0       0     0  100 50366

IAMV.P630232.$IAM.CLUSTER                           2    65429  5040000  280000   20000   40000       0 15761    1  8803
2002.218 J=JFMTESTS S=P630232B P=IAMTVSAM D=VSAMCRT1     567  5000000       0       0       0       0     0  100  8487
         J=JFMTESTS S=P630232B P=IAMTVSAM D=VSAMCRT1   64862  5040000  280000   20000   40000       0 15761    1  8803

IAMV.P630176.CLUSTER                                2    62520    64801   21000   32111       0       0 22750   95   813
2002.218 J=JFMTESTS S=P630176D P=SUBTMNGR D=VSAMCRT1       2        1       0       0       0       0     1    2     2
         J=JFMTESTS S=P630176D P=SUBTMNGR D=VSAMCRT1   62518    64801   21000   32111       0       0 22750   95   813

IAMV.VXA517.CLUSTER                                 2    13613    26892   41310    1385    1237     266   725   14   162
2002.217 J=JFMTESTS S=VXA517A  P=IAMTVSAM D=VSAMCRT1      22    25000       0       0       0       0     0    0   156
         J=JFMTESTS S=VXA517A  P=IAMTVSAM D=VSAMCRT1   13591    26892   41310    1385    1237     266   725   14   162
```

Figure 10: Sample IAMSMFVS Detail Report

The additional fields on the detail report line include:

- **J=** Indicates the job name.
- **S=** Indicates the step name.
- **P=** Indicates the program name.
- **D=** Indicates the DD name.

40.07 IAMSMFVS DATASET SUMMARY REPORT

After the EXCP reports IAMSMFVS produces a DataSet Summary Report. This report contains all of the datasets that were encountered in the SMF data, and retained in the dataset table built by IAMSMFVS. Datasets appear in ascending name sequence. For cataloged VSAM clusters, they are sorted based on the cluster name.

DATASET SUMMARY REPORT

```
IAM400 SMF REPORT/DATA EXTRACT PROGRAM-IAMSMFVS VER 8.0/01P-INNOVATION DATA PROCESSING 2002.218
                              DATA SET SUMMARY REPORT

                             USE   TOTAL                  AVG     MAX  KEY        BLK OR   FRSPC
DATA SET NAME               COUNT  EXCPS DSORG  RECFM LRECL   LRECL LEN RKP   CISIZE  CI% CA%
---- --- ----               -----  ----- -----  ----- -----   ----- --- ---   ------  --- ---
AJM.UPSTREAM.CATALOG            2    373 VSAM
  AJM.UPSTREAM.CATALOG.DATA     2    277 VSAM   VB       50     500  18   0    22528    0   0
  AJM.UPSTREAM.CATALOG.INDEX    2     96 VSAM   NOIMB          2041  18   0     2048
IAMV.IAMTST6.CLUSTER            3     45 IAM    VE-DC   619     700   4   0    11476    0   0
IAMV.P630032.CLUSTER            2   4206 IAM    VE     1000    1000   4   8    11476   10  10
IAMV.P630034.CLUSTER            4   1798 IAM    VE      143     143  24   8    11476    0   0
IAMV.P630042A.CLUSTER           3   2675 IAM    VE     2700    2700   4   8    13682    0  15
IAMV.P630049.CLUSTER            2    328 IAM    VE-DC  2763    4096   8   0    11476    5   5
IAMV.P630232.$IAM.CLUSTER       2  65429 IAM    VE-DC    79    1024   4   8    13682   10  25
IAMV.P630234.$IAM.CLUSTER       2     10 IAM    VE-DC   120    1024   4   8    11476   10  25
IAMV.P630248.CLUSTER            4    374 IAM    FB              128   4   8     4096    0   0
IAMV.P630251.KB.CLUSTER         1      9 IAM    VB     1020    1020   4   8    13682    0   0
IAMV.P630251.MB.CLUSTER         1   1946 IAM    VB     1020    1020   4   8    13682    0   0
IAMV.VIT551.CLUSTER             2 203352 IAM    VE     1020    1020  16   8    11476   10  10
IAMV.VIT552.CLUSTER             2 135086 IAM    VE     1020    1020  64   8    11476   10  10
IAMV.VIT600.CLUSTER             2   1710 IAM    VE     4096    4096   4   0    23476   10  10
IAMV.VXA401.CLUSTER             3      9 IAM    F              2340   4   8    23400   10   0
IAMV.VXA402.CLUSTER             3     19 IAM    VE     4680    2340   4   8    23476   10  10
IAMV.VXA531.CLUSTER             2    748 IAM    VE     1040    1040   4   8    11476   10  10
IAM63.VSAMTEST.DATA            81    243 VSAM
  IAM63.VSAMTEST.DATA.DATA     81    162 VSAM   VB      255     300   8   4    22528   25  10
  IAM63.VSAMTEST.DATA.INDEX    81     81 VSAM   NOIMB           505   8   4      512
SYSPRMM.MASTER                  2    360 VSAM
  SYSPRMM.MASTER.DATA           2    344 VSAM   VB      512    9216  56   0    10240   20  20
  SYSPRMM.MASTER.INDEX          2     16 VSAM   IMBED          2041  56   0     2048
SYSPSMF.CPUB.MAN1               5    900 VSAM
  SYSPSMF.CPUB.MAN1.DATA        5    900 VSAM   VB    22518   32767            22528
```

Figure 11: Sample DataSet Summary Report

40.07 CONTINUED . . .

DATASET SUMMARY REPORT FIELDS The fields that are unique to the DataSet Summary Report are described below. The DataSet Name, Total Use Count, and Total EXCP's are identical to the fields in the EXCP Report.

Field Name	Description
DataSet Name	Specifies the name of the dataset. For cataloged VSAM datasets, this will have the cluster name on the first line, followed by the component names on subsequent lines, which are indented. For uncataloged VSAM datasets, only the component name will be printed.
Use Count	Indicates the number of SMF records processed for this dataset. This will usually be the number of times the dataset was closed during the time interval of the report.
Total EXCPS	The accumulated total EXCP counts from all of the SMF records processed for this dataset. For cataloged VSAM clusters, this number will be the total for all of the components, with the component value given on the subsequent lines.
DSORG	Indicates the dataset organization for the file.
RECFM	Indicates the record format. The following values are possible: **F[B]** – Fixed length record file. For IAM, can only be a Compatible Format datasets. For VSAM, indicates that file was defined with an equal average and maximum record length. **VB** – Variable length record file, applicable to VSAM or Compatible Format IAM datasets. **VE** – Variable length record, IAM Enhanced Format file. **VO** – Variable length record IAM Enhanced Format file with Variable Overflow enabled. **-DC** – IAM Data Compressed file. **IMBED** – VSAM Index Component with an imbedded index. **NOIMB** – VSAM Index Component without an imbedded index.
AVG LRECL	For VSAM data components, indicates the defined average record length. For IAM datasets, indicates the average record length from the file load.
MAX LRECL	Indicates the maximum defined logical record length.
KEY LENGTH	Indicates the defined key length for the dataset.
RKP	Indicates the defined Relative Key Position (or key offset) for the dataset.
BLK or CI SIZE	Indicates the VSAM CI Size or the IAM block size of the dataset.
FREESPACE	Indicates the defined values for CI and CA percent freespace.

40.08 SIZE REPORTS

After the DataSet Summary Report, IAMSMFVS produces size reports. Like the EXCP reports, the Size reports are produced by dataset organization, with only the largest datasets included in each report. The number of datasets included in each size report is from the keyword MAXREPORTS, which defaults to 100. If VSAM datasets are being reported on, that VSAM Size Report will appear first. The IAM Size report, if IAM datasets are being reported on, appears after the VSAM Size Report.

VSAM SIZE REPORT

```
IAM400 SMF REPORT/DATA EXTRACT PROGRAM-IAMSMFVS VER 8.0/01P-INNOVATION DATA PROCESSING  2002.218
                              VSAM SIZE REPORT
                              ALLOC  TOTAL   USE              AVG    MAX    KEY
DATA SET NAME                  TRKS  EXCPS  COUNT  EXTENTS  LRECL  LRECL   LEN  RKP   CISIZE
---- --- ----                 -----  -----  -----  -------  -----  -----   ---  ---   ------
AJM.UPSTREAM.FILEDATA          2705      0      2
  AJM.UPSTREAM.FILEDATA.DATA   2700      0      2        4    840   6165    21    0    22528
  AJM.UPSTREAM.FILEDATA.INDEX     5      0      2        3           505    21    0      512
SYSPSMF.CPUB.MAN1               900    900      5
  SYSPSMF.CPUB.MAN1.DATA        900    900      5        1  22518  32767     0    0    22528
SID.UPSTREAM.FILEDATA           345      0      2
  SID.UPSTREAM.FILEDATA.DATA    330      0      2        3    840   6165    21    0    22528
  SID.UPSTREAM.FILEDATA.INDEX    15      0      2        1          4089    21    0     4096
SYSPRMM.MASTER                  286    360      2
  SYSPRMM.MASTER.DATA           285    344      2        1    512   9216    56    0    10240
  SYSPRMM.MASTER.INDEX            1     16      2        1          2041    56    0     2048
USTEST.UPSTREAM.FILEDATA        151      4      3
  USTEST.UPSTREAM.FILEDATA.DATA 150      1      3        1    840   6165    21    0    22528
  USTEST.UPSTREAM.FILEDATA.INDEX  1      3      3        1          4089    21    0     4096
SYSPSMF.CPUB.MAN2               150    600      2
  SYSPSMF.CPUB.MAN2.DATA        150    600      2        1  22518  32767     0    0    22528
AJM.UPSTREAM.CATALOG             91    373      2
  AJM.UPSTREAM.CATALOG.DATA      90    277      2        2     50    500    18    0    22528
  AJM.UPSTREAM.CATALOG.INDEX      1     96      2        1          2041    18    0     2048
IAM63.VSAMTEST.DATA              46    243     81
  IAM63.VSAMTEST.DATA.DATA       45    162     81        1    255    300     8    4    22528
  IAM63.VSAMTEST.DATA.INDEX       1     81     81        1           505     8    4      512
```

Figure 12: Example IAMSMFVS VSAM Size Report

IAM SIZE REPORT

```
IAM400 SMF REPORT/DATA EXTRACT PROGRAM-IAMSMFVS VER 8.0/01P-INNOVATION DATA PROCESSING
                                           IAM SIZE REPORT
                         TRACKS   TOTAL   USE     TOTAL     INDEPENDENT OVERFLOW    PRIME
DATA SET NAME              USED   EXCPS COUNT   RECORDS   MAX REC  USE REC  % USE    EXT   CI%
---- --- ----            ------   ----- -----   -------   -------  -------  -----  -----   ---
IAMV.P630175A.CLUSTER    100048    6713     3   2000020       660       10      1      0     0
IAMV.VIT552.CLUSTER       50366  135086     2   2000000         0        0      0      0    10
IAMV.VIT551.CLUSTER       50086  203352     2   2000000         0        0      0      0    10
IAMV.P630042A.CLUSTER     39238    2675     3    784360     54985       30      0      0     0
IAMV.P630119.CLUSTER      33016   13704     2   1716000         0        0      0      0     0
IAMV.P630180.CLUSTER      32517    2204     3    650020      9670       10      0      0     0
IAMV.P630251.MB.CLUSTER   14586    1946     1    720000     38038        0      0      4     0
IAMV.P630115.CLUSTER      12305     822     1    565800         0        0      0      0     0
IAMV.P630159A.CLUSTER      9007     621     3    360020     72340       10      0      0     0
IAMV.P630147.CLUSTER       7955    1092     2    318000         0        0      0      0     0
IAMV.P630150.CLUSTER       7505   24506     2    240000         0        0      0      0    15
IAMV.P630132.CLUSTER       5366     755     4     32139         0        0      0      0    10
IAMV.P630131.CLUSTER       2887   10922     3    215000     20740        0      0  10000     0
IAMV.P630183.CLUSTER       2504     415     4    200000         0        0      0      0     0
IAMV.P630075.CLUSTER       1693    5521     3     50001     37556     7998     21   1374    10
IAMV.P630076.CLUSTER       1693    7846     4     50001     37556     7998     21   1374    10
IAMV.P630187.CLUSTER       1020    4788     3     72000    144984    24000     16      0     0
```

Figure 13: Example IAMSMFVS IAM Size Report

40.08 CONTINUED . . .

SIZE REPORT FIELD DESCRIPTIONS

Field Name	Description
DataSet Name	Specifies the name of the dataset. For cataloged VSAM datasets, this will have the cluster name on the first line, followed by the component names on subsequent lines, which are indented. For uncataloged VSAM datasets, only the component name will be printed.
Alloc Tracks	For VSAM components, the maximum number of tracks allocated to the component.
Tracks Used	For IAM datasets, indicates the maximum amount of DASD space used for this file.
Total EXCPS	The accumulated total EXCP counts from all of the SMF records processed for this dataset. For cataloged VSAM clusters, this number will be the total for all of the components, with the component value given on the subsequent lines.
Use Count	Indicates the number of SMF records processed for this dataset. This will usually be the number of times the dataset was closed during the time interval of the report.
Extents	For VSAM clusters, indicates maximum number of DASD extents.
AVG LRECL	For VSAM data components, indicates the defined average record length.
MAX LRECL	For VSAM, indicates the maximum defined logical record length.
KEY LENGTH	For VSAM, indicates the defined key length for the dataset.
RKP	For VSAM, indicates the defined Relative Key Position (or key offset) for the dataset.
CI SIZE	For VSAM, indicates the VSAM CI Size which is the amount of data transferred per physical I/O.
Total Records	For IAM, contains the maximum records that were in the file, or if CURRENT was specified, the current number of records in the file.
Max Rec	For Enhanced Format files, without an overflow override, the estimated number of records that can fit within the currently allocated extents. For Enhanced Format files with an Overflow override, this will be that value. For Compatible format files, the maximum size of the overflow area.
Use Rec	The maximum records in overflow, or if CURRENT was specified on the REPORT card, in which case it is the most recent value.
% USE	The percentage of the overflow area that has been used.
Prime Ext	The number of Prime Extension Blocks.
CI%	For IAM datasets, the size of Integrated Overflow, normally specified by CI Freespace.

THIS PAGE WAS INTENTIONALLY LEFT BLANK

41.01 SMF JOB RESOURCE USAGE ANALYSIS OVERVIEW

OVERVIEW The IAM product includes a specialized SMF reporting program called IAMSMF. This program is intended to aid customers in converting VSAM datasets to IAM and in using IAM datasets in their installation. Customers can use IAMSMF to:

1. Aid in conversion of datasets from VSAM to IAM, by providing the capability to determine what jobs and job steps are defining the VSAM datasets, as well as processing the selected datasets.
2. Provide reports on SMF job statistics to help evaluate the benefit of IAM. You can obtain reports that include CPU time, EXCP's, and elapsed time for jobs using VSAM datasets. Then, compare the jobs based on those statistics to see the improvements realized after converting the VSAM datasets to IAM.
3. Provide detailed reports on IAM dataset activity by producing the IAMINFO reports from the IAM SMF records.
4. Find out what jobs or users are accessing datasets.
5. Print or copy selected SMF records.

FUNCTIONAL SUMMARY The IAMSMF program has been designed to give users the ability to extract data and statistics from Systems Management Facility (SMF) records. The functions include dataset query, printing IAMINFO reports, job reporting, record copy, and record print. IAMSMF is a batch program, which based upon simple control statements, will extract resource utilization information from SMF job/step and dataset records. Information can be selected directly from the active system SMF datasets, from SMF format archival files, or from extract files created by the COPY function of the program itself.

COMMAND SUMMARY IAMSMF has the following commands:

COPY: Copy selected SMF records to a sequential file creating a subset of the data for later use.

IAMINFO: Produce IAMINFO reports from IAM generated SMF records.

PRINT: Print selected SMF records in dump (Hexadecimal) format.

QUERY: Produces a dataset oriented report, to find out what job steps use the specified datasets, and how they are used. Can be requested for specific dataset name(s) or by dataset name prefixes (dataset groups).

REPORT: Produces a job oriented report, with a break out or resource and dataset utilization by job step. The report is produced in chronological order from the available SMF data. Can be requested for a specific Jobname(s) or by Job Group Name(s).

41.02 JCL REQUERIMENTS FOR IAMSMF

The JCL statements required to execute IAMSMF are as follows:

EXECUTE STATEMENT Specifies the name of the IAM SMF Analysis program: IAMSMF. The region size used by IAMSMF is at least 512K. A typical execute statement would be:

//SMFQUERY EXEC PGM=IAMSMF,REGION=1M

DD STATEMENTS The following table identifies the required DD statements.

DD Name	Description
STEPLIB or JOBLIB	It is recommended that the IAM program load library be included in the system link list. If it is not you must include a STEPLIB/JOBLIB DD statement specifying the IAM load library that contains the IAMSMF program.
SYSPRINT	Specifies where the IAMSMF control statements, messages and reports are to be printed. Usually a SYSOUT dataset.
SYSMF	Specifies the SMF source dataset. This DD statement may point to one or more of the active SMF datasets on disk (ex: SYS1.MANx), an SMF format archival file, or a sequential file produced by the COPY function of this program. An alternate DD name may be specified via a control statement.
SYSUT2	Specifies the output dataset for a COPY operation. Usually a sequential file on tape or disk. An alternate DD name may be specified via a control statement.
SYSIN	Specifies the input control statement card image dataset. Usually a DD * dataset.

EXAMPLE A: BASIC IAMSMF JCL EXAMPLE An example of the JCL to run IAMSMF is shown below. The SYSUT2 DD card is required for a COPY command, otherwise it is not needed.

```
//IAMSMF    EXEC  PGM=IAMSMF,REGION=1M
//STEPLIB    DD   DISP=SHR,DSN=iam.load.lib
//SYSPRINT   DD   SYSOUT=*
//SYSMF      DD   DISP=SHR,DSN=smf.data.set
//SYSUT2     DD   DSN=my.smf.records,UNIT=SYSDA,DISP=(,CATLG),
//                SPACE=(CYL,(10,1))
//SYSIN      DD    *

       IAMSMF Control cards are inserted here

/*
```

Figure 14: Example of Basic JCL to run IAMSMF (EX4102A)

41.03 IAMSMF - COPY COMMAND

COPY COMMAND

The COPY command is used to copy selected SMF records to a sequential file from either a history file (RECFM=VBS) or an active SMF data recording file. The COPY command can be used to create a subset of the full SMF data so that reports can be generated at a later point in time for comparisons. You might want to save a subset of the records, such as all of the IAM SMF records, in one place and/or for a longer period of time than the normal SMF data is kept. Another use of copy is to aid in creating reports to find out particular information. For example, if you want to find all the jobs that are defining VSAM or IAM datasets, but did not want them intermixed with other file activity, first copy the record types 61 and 63, which are the VSAM Define SMF records. Then run the IAMSMF Query command from the data subset.

COPY	
[ALLRECORDS]	**[,JOBNAMES=**jobname**]**
[,CHECKLENGTH]	**[,MAXRECLENGTH=**nnnn **]**
[,ERRORPRINT]	**[,PRTLENGTH=**nnn **]**
[,FROMDATE=yyyyddd **]**	**[,RECSIZE(rrr)=**nnnn **]**
[,FROMDDNAME=ddname **]**	**[,RECTYPE=**nnn **]**
[,GROUPNAMES=jobname **]**	**[,TODATE=**yyyyddd **]**
[,IAMRECORDS]	**[,TODDNAME=**ddname **]**

Figure 15: IAMSMF COPY Command Operands

41.03 CONTINUED . . .

COPY COMMAND OPERANDS

Operand	Description
ALLRECORDS	Specifies that the SMF record type does not participate in record selection. The default is deferred to the operand RECTYPE. **NOTE:** This operand conflicts with the operand RECTYPE.
CHECKLENGTH	Specifies that SMF data records are to be validated against a table of minimum record lengths. If the user has modified the minimum record length or is executing in a non-compatible system, the correct record lengths may be specified by the RECSIZE operand. The default is SMF record length is not validated.
ERRORPRINT	Specifies that any SMF data record that causes an error during processing or fails length verification is to be printed. The default is records in error are not printed.
FROMDATE=	Specifies the lower date limit of the SMF records that are to be copied. The date has the format of yyyyddd or yyddd. The shorter format assumes a prefix of 19. The default, if the FROMDATE and/or TODATE operands are not specified, is that the date of the SMF record will not participate in the selection criteria.
FROMDDNAME=	Specifies the DDNAME of the SMF file to be used as input to IAMSMF. The default input DDNAME is SYSMF.
GROUPNAMES=	Specifies that only records having a jobname which begin with the specified character string(s) will be copied. This operand specifies a partial jobname from 1 to 8 characters in length. Up to 50 job groups and/or names may be specified for a single command if entered as follows: **GROUPNAMES=**(jobname1,...,jobnamex) The default, if neither the GROUPNAMES nor JOBNAMES operand is specified, is that the jobname will not participate in SMF record selection.
IAMRECORDS	Specifies that SMF records generated by IAM are to be selected. The default is deferred to the operand RECTYPE. **NOTE:** This operand conflicts with the operands RECTYPE and ALLRECORDS.
JOBNAMES=	Specifies that only records having a jobname which match the jobname(s) specified will be copied. This operand specifies a complete jobname from 1 to 8 characters in length. Up to 50 job names may be specified in a single command if entered as follows: **JOBNAMES=**(jobname1,...,jobnamex) The default, if neither the GROUPNAMES nor JOBNAMES operand is specified, is that the jobname will not participate in SMF record selection.
MAXRECLENGTH=	Specifies the largest SMF record that the program will process. The number may be any value from 16384 to 65536, inclusive. The default is 16384 bytes.

41.03 CONTINUED . . .

PRTLENGTH= Limits the amount of data to be printed if the ERRORPRINT keyword is specified. The number may be any value from 32 to 65536, inclusive.

The default is 32768 bytes.

RECSIZE(rrr)= Establishes the minimum length of the SMF record type 'rrr' as the value 'nnnn'.

The default minimum record lengths for system generated SMF records are documented in the IBM Systems Management Facilities manual.

RECTYPE= Identifies the specific record type(s) to be copied. Up to 50 record types may be specified for a single command if entered as follows:

RECTYPE=(rrr,...,rrr)

The SMF record types which will be copied by default (if ALLRECORDS is not specified) are as follows:

4 – Step termination

5 – Job termination

14 – NON-VSAM Dataset CLOSEd (input)

15 – NON-VSAM Dataset CLOSEd (output/update)

20 – Job initiation

30 – Common Address Space Work Record

34 – TSO session termination

64 – VSAM Dataset CLOSEd

NOTE: This operand conflicts with the operand ALLRECORDS.

TODATE= Specifies the upper date limit of the SMF records that are to be copied. The format is yyyyddd or yyddd. The shorter format assumes a prefix of 19.

The default, if the FROMDATE and/or TODATE operands are not specified, is that the date of the SMF record will not participate in the selection criteria.

TODDNAME= Specifies the DDNAME of the output file for the copied SMF records.

The default output DDNAME is SYSUT2.

41.03 CONTINUED . . .

EXAMPLE A: COPY COMMAND USAGE

This example shows how to create a subset of SMF records that can be used for subsequent input to the IAMSMFVS analysis program. The record types being copied include the Type 30 for the step termination, the Type 64 for VSAM datasets, and the Type 201, which is the IAM SMF record type specified in the IAM Global Options Table. The output dataset created by this job can be used as an input dataset for IAMSMFVS. The FROMDATE and TODATE keywords are specified indicating to copy data for the month of November 1997. The SYSUT2 DD specifies a DISP=MOD, indicating that records are being added to an already existing file, which contains previously extracted data from prior months. A MAXRECLENGTH keyword was specified to insure that there would be no records dropped.

```
//IAMSMF   EXEC  PGM=IAMSMF,REGION=1M
//STEPLIB  DD    DISP=SHR,DSN=iam.load.lib
//SYSPRINT DD    SYSOUT=*
//SYSMF    DD    DISP=SHR,DSN=smf.data.set
//SYSUT2   DD    DSN=my.smf.records,DISP=(MOD,KEEP,KEEP)
//SYSIN    DD     *
      COPY RECTYPE=(30,64,201),FROMDATE=2002.200,TODATE=2002.217,
      MAXRECLENGTH=32768
/*
```

Figure 16: Example of IAMSMF COPY Command (EX4103A)

SAMPLE IAMSMF COPY OUTPUT

Below is sample output from running the above job. The IAM601 message indicates that 5,018 records were copied to the output dataset, and that no records were dropped because of too long of a record length.

```
IAM400 SMF REPORT/DATA EXTRACT PROGRAM-IAMSMF VER 8.0/01P INNOVATION DATA PROCESSING DATE 2002.218
IAM303   CARD IMAGE - *   COPY  RECTYPE=(30,64,201),FROMDATE=2002.200,TODATE=2002.217,         *
IAM303   CARD IMAGE - *         MAXRECLENGTH=32768                                            *
IAM491   RECORD COPY      FUNCTION STARTED - 11.01.23

IAM601   SMF RECORDS -- READ.....65003  USED......5018  DROPPED.........0
IAM492   RECORD COPY      FUNCTION ENDED   - 11.01.47 - CONDITION CODE 000

IAM499   IAMSMF(8.0/01P ) PROCESSING COMPLETED
```

Figure 17: Sample Output from IAMSMF COPY Command

41.04 IAMSMF - IAMINFO COMMAND

IAMINFO COMMAND

The IAMINFO command is used to produce the detailed IAMINFO reports from IAM generated SMF records. IAM will normally produce an IAMINFO report each time an IAM dataset is closed if the job step has an IAMINFO DD card. By collecting the IAM SMF records, you can still get the IAMINFO reports without changing the JCL in every job using IAM datasets. Plus, this offers the advantage of being able to obtain the IAMINFO reports only on an as needed or desired basis, or if the original job output has been discarded.

To obtain a summarization of IAM dataset activity, use the IAMSMFVS program. That program provides three distinct reports, consisting of one line per dataset rather than multiple full page reports. You could use IAMSMFVS to obtain the summary information, and then use the IAMSMF IAMINFO command to get detailed information about specific files.

Each IAMINFO report is one page in length, providing detailed information about the dataset and it's use. An IAMNFO report is generated by IAMSMF as one per dataset per job step. Optionally, IAMINFO reports can be printed by IAMSMF for each time the dataset is closed, or once per dataset per job. From the IAMINFO reports, you can tell how a program is accessing the dataset, whether or not more buffers could help improve performance, and if the dataset needs to be reorganized.

The IAMINFO command offers a wide variety of selection criteria to provide a lot of flexibility in the reports that end up being printed.

IAMINFO	
[ATTRIBUTE=ccccc**]**	**[,MAXSTACK=**nnnnn**]**
[,BLOCKS=nnnnnn**]**	**[,MERGE=**ccccc**]**
[,CYLS=nnnnn**]**	**[,NUMCMDS=**nnnnn**]**
[,DSGROUPS=dsname**]**	**[,NUMRECS=**nnnnn**]**
[,DSNAMES=dsname**]**	**[,OFULL=**nn**]**
[,DYNRET=nnnnn**]**	**[,ORECS=**nnnnn**]**
[,EXCP=nnnnn**]**	**[,PEBLKS=**nnnnn**]**
[,FROMDATE=yyyyddd**]**	**[,PEFULL=**nn**]**
[,FROMDDNAME=ddname**]**	**[,RECTYPE=**nnn**]**
[,GROUPNAMES=jobname**]**	**[,TODATE=**yyyyddd**]**
[,JOBNAMES=jobname**]**	**[,TRACKS=**nnnnn**]**
[,LRECL=nnnnn**]**	**[,TYPECMD[E]=(**a,b,c**)]**
[,SHAREOPTION=n**]**	

Figure 18: IAMSMF IAMINFO Command Operands

41.04 CONTINUED . . .

IAMINFO COMMAND OPERANDS

Operand	Description
ATTRIBUTE=	Identifies which attributes an IAM file must have to participate in record selection. Valid values are: **DATACOMP** – The IAM file contains compressed data. **KEYCOMP** – The IAM file contains compressed keys. **INCPEBLK** – IAM found that a record that could have gone into PRIME EXTENSION, but was placed into INDEPENDENT OVERFLOW because the PRIME EXTENSION was full. Redefining this file with a larger PRIME EXTENSION may improve performance. An IAM371 message was issued if an IAMINFO DD was present during the job's execution. **MOREBUFFER** – IAM found that additional buffers could have been used to improve performance, however the MAXBUF or BUFNO option prevented IAM from acquiring more buffers. An IAM368 message was issued if an IAMINFO DD was present during the job's execution. **NOCORE** – There was not enough storage available in the region for IAM to acquire additional buffers. An IAM367 message was issued if an IAMINFO DD was present during the job's execution. **REORG** – IAM files that have received warning messages recommending that the file be reorganized. By default, the dataset attributes do not participate in the selection criteria. This operand supports the following logical operators: =, ¬=.
BLOCKS=	Specifies the size, in blocks, of the files to be selected. By default, the size of the IAM dataset does not participate in the selection criteria. This operand supports the following logical operators: =, ¬=, >, > =, <, < =
CYLS=	Specifies the size, in cylinders, of the files to be selected. By default, the size of the IAM dataset does not participate in the selection criteria. This operand supports the following logical operators: =, ¬ =, >, > =, <, < =
DSGROUPS=	Specifies that only records having a dataset name which begin with the given character string(s) will be processed. This operand specifies a partial dataset name from 1 to 44 characters in length. Up to 50 dataset groups may be specified for a single command if entered as follows: **DSGROUPS**=(dsname1,...,dsnamex) By default, the name of the IAM dataset does not participate in the selection criteria.

41.04 CONTINUED . . .

DSNAMES= Specifies that only records having a dataset name which match the dataset name(s) specified will be processed. This operand specifies a complete dataset name from 1 to 44 characters in length. Up to 50 dataset names may be specified for a single command if entered as follows:

DSNAMES=(dsname,...,dsname)

By default, the name of the IAM dataset does not participate in the selection criteria.

DYNRET= Establishes the limit of record retrievals from IAM's Dynamic Table.

By default, the number of dynamic retrievals does not participate in the selection criteria.

This operand supports the following logical operators: =, ¬ =, >, > =, <, < =

EXCP= Establishes the limit of EXCPs (physical read and writes) to the IAM file.

By default, the number of EXCPs does not participate in the selection criteria.

This operand supports the following logical operators: =, ¬ =, >, > =, <, < =

FROMDATE= Specifies the lower date limit of the SMF records that are to be copied.

The default, if the FROMDATE and/or TODATE operands are not specified, is that the date of the SMF record will not participate in the selection criteria.

FROMDDNAME= Specifies the DDNAME of the SMF file to be used as input to IAMSMF.

The default input DDNAME is SYSMF.

GROUPNAMES= Specifies that only those records having a job name which begin with the specified character string(s) will be selected. This operand specifies a partial job name from 1 to 8 characters in length. Up to 50 job groups and/or names may be specified for a single command if entered as follows:

GROUPNAMES=(jobname1,...,jobnamex)

The default, if neither the GROUPNAMES nor JOBNAMES operand is specified, is that the job name will not participate in SMF record selection.

JOBNAMES= Specifies that only records having a job name which match the jobname(s) specified will be selected. This operand specifies a complete job name from 1 to 8 characters in length. Up to 50 job names may be specified in a single command if entered as follows:

JOBNAMES=(jobname1,...,jobnamex)

The default, if neither the GROUPNAMES nor JOBNAMES operand is specified, is that the job name will not participate in SMF record selection.

LRECL= Establishes the limit for the length of records (maximum length for variable) within an IAM file.

By default, the record length does not participate in the selection criteria.

This operand supports the following logical operators: =, ¬ =, >, > =, <, < =

MAXSTACK= Specifies the maximum number of compressed SMF records that can be retained in storage. The number can be any value from 100 to 50000, inclusive.

The default is 2500 records.

41.04 CONTINUED . . .

MERGE= Identifies how IAMSMF will process multiple SMF records created for the same dataset by the same job. Valid values are:

JOB: – All SMF records referencing the same IAM dataset within a job will be merged into a single report.

NO: – No IAM SMF records are to be merged. An IAMINFO report will be generated for each IAM SMF record found.

STEP: – All SMF records referencing the same IAM dataset within a job step (i.e. multiple OPEN/CLOSE) will be merged into a single report.

The default is STEP.

NUMCMDS= Establishes the limit of the total number of commands issued against the IAM file. When used in conjunction with the TYPECMD operand, the selection will be limited to those files having a command count of that type.

By default, the number of commands issued against the IAM file does not participate in the selection criteria.

This operand supports the following logical operators: =, ¬ =, >, > =, <, < =

OFULL= Establishes the percent of overflow used limit within an IAM file.

By default, the percent of overflow used does not participate in the selection criteria.

This operand supports the following logical operators: =, ¬ =, >, > =, <, < =

ORECS= Establishes the number of overflow records limit within an IAM file.

By default, the number of overflow records does not participate in the selection criteria.

This operand supports the following logical operators: =, ¬ =, >, > =, <, < =

PEBLKS= Establishes the number of prime extension blocks limit within an IAM file.

By default, the number of prime extension blocks does not participate in the selection criteria.

This operand supports the following logical operators: =, ¬ =, >, > =, <, < =

PEFULL= Establishes the percent of prime extension blocks used limit within an IAM file.

By default, the percent of prime extension blocks used does not participate in the selection criteria.

This operand supports the following logical operators: =, ¬ =, >, > =, <, < =

RECTYPE= Identifies the record type of the IAM generated SMF record.

The default record type is that which is defined in the IAM option table. Refer to section 91 for further documentation on the IAM option table.

SHAREOPTION= Specifies the cross region share options of the files to be selected. Valid values are 1,2,3, or 4.

By default, the share options do not participate in the selection criteria.

This operand supports the following logical operators: =, ¬ =, >, > =, <, < =

TODATE= Specifies the upper date limit of the SMF records that are to be copied.

The default, if the FROMDATE and/or TODATE operands are not specified, is that the date of the SMF record will not participate in the selection criteria.

41.04 CONTINUED . . .

TRACKS= Specifies the size, in tracks, of the files to be selected.

By default, the size of the IAM dataset does not participate in the selection criteria.

This operand supports the following logical operators: =, ¬ =, >, > =, <, < =

TYPECMD= Identifies the type of command(s) that must have been issued against Compatible Format IAM files.

If NUMCMD is also specified, the SMF record will be selected if that number of commands of that type was issued.

If NUMCMD is not specified, the SMF record will be selected if ANY command of that type was issued.

Up to 16 command types may be specified if entered as follows:

TYPECMD=(c,c,...,c)

The following values (command types which correspond to those generated by IAMINFO) are supported:

R – Read, **W** – Write, **F** – Getnext,
A – Add, **D** – Delete, **G** – Get,
S – Start Key Equal, **K** – Start KC or Point, **N** – Read or Getnext,
P – Get previous, **U** – Read for Update, **E** – Stastics,
C – Close, **O** – Open, **I** – Information, or
X – Flush

By default, the type of commands issued against the IAM file does not participate in the selection criteria.

TYPECMDE= Identifies the type of command(s) that must have been issued against Enhanced Format IAM files.

If NUMCMD is also specified, the SMF record will be selected if that number of commands of that type was issued.

If NUMCMD is not specified, the SMF record will be selected if ANY command of that type was issued.

Up to 16 command types may be specified if entered as follows:

TYPECMDE=(c,c,...,c)

The following values (command types which correspond to those generated by IAMINFO) are supported:

A – Add, **C** – Close, **E** – Erase,
F – Skip Sequential Get, **G** – Get Sequential, **I** – IAM Statistics,
K – Point, **L** – Record Length Change,
N – Point Key Greater or Equal, or Generic, **O** – Open,
P – Get Previous (Backwards Get),
R – Random Read with Key Equal,
S – Random Read with Key Greater or Equal or Generic, **T** – Close Type=T,
U – Put/ Write for Update, **V** – Verify, or
X – Buffer Flush Request.

By default, the type of commands issued against the IAM file does not participate in the selection criteria.

41.04 CONTINUED . . .

IAMINFO OUTPUT

The IAMINFO output produced by IAMSMF is quite similar to the normal IAMINFO report. The only change has been in the heading messages. When produced by IAMSMF, the IAMINFO reports do not have the IAM360 or IAM361 heading messages. Rather, they have an IAM370 JOB CHARACTERISTICS section. A sample of that portion of the IAMINFO report is shown below. For a complete description of the IAMINFO report, please refer to Section 10.70 IAM Reports.

```
IAM370   JOB CHARACTERISTICS -
         JOB NAME ----------------=    KSD270    -    STEP NAME ----------=    KSD270B
         PROGRAM NAME ------------=   IAMTVSAM   -    FUNCTION -----------=    FILE CREATION
         DDNAME ------------------=   VSAMCRT1   -    DSNAME = IAMV.KSD270.CLUSTER
         DATE OPENED -------------=   2002.218   -    TIME OPENED ---------=   15:05:11
         DATE CLOSED -------------=   2002.218   -    TIME CLOSED ---------=   15:05:11
```

Figure 19: Sample of IAM370 IAMINFO Output

JOB CHARACTERISTICS FIELD DESCRIPTION

Field Name	**Description of Field**
JOB NAME	Indicates the name of the job that has processed this IAM dataset.
STEP NAME	Indicates the name of the job step that has processed this IAM dataset.
PROGRAM NAME	The name of the program that issued the OPEN for the IAM dataset.
FUNCTION	The function being performed. Possible values are: **FILE CREATION** – Indicates a file load process. **UPDATE PROCESSING** – Indicates that the file was opened for update. **INPUT PROCESSING** – Indicates that the file was opened for input processing only.
DDNAME	Indicates the DD name used to OPEN the file.
DSNAME	Indicates the IAM dataset name.
DATE OPENED	The date that the dataset was opened.
TIME OPENED	The time of the OPEN.
DATE CLOSED	Indicates the date that the dataset was closed.
TIME CLOSED	Indicates the time of the CLOSE.

Figure 20: IAM370 Job Characteristics Field Descriptions

41.04 CONTINUED . . .

EXAMPLE A: REQUESTING A SPECIFIC JOB AND DATASET

In this first example of running IAMINFO from IAMSMF, we are after a specific IAMINFO report. The JOBNAME indicates the name of the job, the DSNAME indicates the name of the dataset, and the FROMDATE and TODATE indicate the date that the job was run. MERGE=NO is also specified, so the IAMINFO reports come out for each close, rather than being merged by job step.

```
//IAMSMF   EXEC  PGM=IAMSMF,REGION=1M
//STEPLIB  DD    DISP=SHR,DSN=iam.load.lib
//SYSPRINT DD    SYSOUT=*
//SYSMF    DD    DISP=SHR,DSN=smf.data.set
//SYSIN    DD    *
      IAMINFO DSNAME=IAMV.KSD270.CLUSTER,JOBNAME=KSD270,
      MERGE=NO,FROMDATE=2002218,TODATE=2002218
/*
```

Figure 21: Example 1 of IAMSMF IAMINFO Command (EX4104A)

EXAMPLE B: IAMINFO REPORTS FOR ALL DATASETS FOR A PARTICULAR JOB

This next example is requesting all of the IAMINFO reports for jobs beginning with the specified GROUPNAME, that ran on the specified date.

```
//IAMSMF   EXEC  PGM=IAMSMF,REGION=1M
//STEPLIB  DD    DISP=SHR,DSN=iam.load.lib
//SYSPRINT DD    SYSOUT=*
//SYSMF    DD    DISP=SHR,DSN=smf.data.set
//SYSIN    DD    *
      IAMINFO GROUPNAME=KSD2,MERGE=NO,
      FROMDATE=2002218,TODATE=2002218
/*
```

Figure 22: Example 2 of an IAMSMF IAMINFO Command (EX4104B)

EXAMPLE C: USING SELECTIVE COMPARISONS

In this next example, two IAMINFO commands are being issued. To reduce processing time, the IAM SMF records are first copied to a temporary dataset, which is then used for input to the IAMINFO commands. The first IAMINFO is requesting all files opened under the various CICS regions that could have used more buffers to be printed. As part of the screening, we are only concerned about those files that have relatively heavy I/O, so EXCP>10000 are included. From this, a determination can be made if the maximum buffer number really should be raised.

The second IAMINFO command will find jobs that have done a large quantity of inserts. This is done by specifying TYPECMD=A, which means ADD requests, and NUMCMDS>10000, which indicates more than 10000 adds.

```
//IAMSMF   EXEC  PGM=IAMSMF,REGION=1M
//STEPLIB  DD    DISP=SHR,DSN=iam.load.lib
//SYSPRINT DD    SYSOUT=*
//SYSMF    DD    DISP=SHR,DSN=smf.data.set
//SYSUT2   DD    UNIT=SYSDA,SPACE=(CYL,(10,5))
//SYSIN    DD    *
    COPY IAMRECORDS
    IAMINFO  GROUPNAMES=CICS,ATTR=MOREBUFFER,EXCP>10000,
    FROMDDNAME=SYSUT2
    IAMINFO  NUMCMDS>10000,TYPECMD=A,FROMDD=SYSUT2
/*
```

Figure 23: Example 3 of IAMSMF IAMINFO Command (EX4104C)

41.05 IAMSMF - PRINT COMMAND

PRINT COMMAND

The PRINT command is used to print selected SMF records from either a history file (RECFM=VBS) or an active SMF data recording file. The primary use of the PRINT command is typically to diagnose unexpected results from the IAMSMF or IAMSMFVS analysis programs. The PRINT command is useful for verifying that particular types of SMF records are actually being collected, and also to manually verify the format of the record.

PRINT	
[[ALLRECORDS]	**[,MAXPRINT=** nnn**]**
[,CHECKLENGTH]	**[,MAXRECLENGTH=** nnnn **]**
[,ERRORPRINT]	**[,PRTLENGTH=** nnn **]**
[,FROMDATE= yyyyddd **]**	**[,RECSIZE(rrr)=** nnnnn **]**
[,FROMDDNAME= ddname**]**	**[,RECTYPE=** nnn **]**
[,GROUPNAMES= jobname**]**	**[,TODATE=** yyyyddd **]**
[,JOBNAMES= jobnames **]**	

Figure 24: IAMSMF Print Command Operands

PRINT COMMAND OPERANDS

Operand	Description
ALLRECORDS	Specifies that the SMF record type does not participate in record selection. The default is deferred to the operand RECTYPE. **NOTE:** This operand conflicts with the operand RECTYPE.
CHECKLENGTH	Specifies that SMF data records are to be validated against a table of minimum record lengths. If the user has modified the minimum record length or is executing in a non-compatible system, the correct record lengths may be specified by the RECSIZE operand. By default, the length of an SMF record is not validated.
ERRORPRINT	Specifies that any SMF data record that causes an error during processing or fails length verification is to be printed. The default is records in error are not printed.
FROMDATE=	Specifies the lower date limit of the SMF records that are to be printed. The default is that there is no lower limit on the dates of the SMF records to eligible to be printed.
FROMDDNAME=	Specifies the DDNAME of the SMF file to be used as input to IAMSMF. The default input DDNAME is SYSMF.

41.05 CONTINUED . . .

GROUPNAMES= Specifies that only the records having a jobname which begin with the specified character string(s) will be printed. This operand specifies a partial jobname from 1 to 8 characters in length. Up to 50 job groups and/or names may be specified for a single command if entered as follows:

GROUPNAMES=(jobname1,...,jobnamex)

The default, if neither the GROUPNAMES nor JOBNAMES operand is specified, is that the jobname will not participate in SMF record selection.

JOBNAMES= Specifies that only the records having a jobname which match the jobname(s) specified will be printed. This operand specifies a complete jobname from 1 to 8 characters in length. Up to 50 jobnames may be specified in a single command if entered as follows:

JOBNAMES=(jobname1,...,jobnamex)

The default, if neither the GROUPNAMES nor JOBNAMES operand is specified, is that the jobname will not participate in SMF record selection.

MAXPRINT= Specifies the maximum number of records the program will print. The number may be any value from 1 to 65536, inclusive.

The default is 20 records.

MAXRECLENGTH= Specifies the largest SMF record that the program will process. The number may be any value from 16384 to 65536, inclusive.

The default is 16384 bytes.

PRTLENGTH= Limits the amount of data to be printed if ERRORPRINT is indicated. The number may be any value from 32 to 65536, inclusive.

The default is 32768 bytes.

RECSIZE(rrr)= Establishes the minimum length of the SMF record type 'rrr' as the value 'nnnn'.

The default minimum record lengths for system generated SMF records are documented in the IBM Systems Management Facilities manual.

RECTYPE= Identifies the specific record type(s) to be copied. Up to 50 record types may be specified for a single command if entered as follows: RECTYPE=(rrr,...,rrr)

The SMF record types which will be copied by default are as follows:

4 – Step termination

5 – Job termination

14 – NON-VSAM Dataset CLOSEd (input)

15 – NON-VSAM Dataset CLOSEd (output/update)

20 – Job initiation

30 – Common Address Space Work Record

34 – TSO session termination

64 – VSAM Dataset CLOSEd

NOTE: This operand can NOT be specified if ALLRECORDS has been specified.

TODATE= Specifies the upper date limit of the SMF records that are to be printed.

The default, is that the date of the SMF record will not participate in the selection criteria.

41.05 CONTINUED . . .

EXAMPLE A: PRINT

An example of running the IAMSMF PRINT command is shown below, followed by sample output. In this example, to make sure that SMF Type 64 records (VSAM Close) are being produced, the PRINT command specifies RECTYPE=64, and this request is being limited to jobs beginning with a U, as specified by the GROUPNAME=U.

```
//IAMSMF    EXEC   PGM=IAMSMF,REGION=1M
//STEPLIB   DD     DISP=SHR,DSN=iam.load.lib
//SYSPRINT  DD     SYSOUT=*
//SYSMF     DD     DISP=SHR,DSN=smf.data.set
//SYSIN     DD     *
       PRINT   RECTYPE=(64),GROUPNAME=U
/*
```

Figure 25: Example IAMSMF PRINT Command JCL (EX4105A)

```
IAM400 SMF REPORT/DATA EXTRACT PROGRAM-IAMSMF VER 8.0/01P INNOVATION DATA PROCESSING DATE-2002.218
IAM303   CARD IMAGE - *   PRINT RECTYPE=(64),GROUPNAME=U                                *
IAM491   RECORD PRINT     FUNCTION STARTED - 16.27.23

RECORD TYPE...64  LENGTH....446  SYSID..OS10  DATE...1997.322  TIME...15.19.02
000000 01BE0000 1E400054 23DF0097 322FD6E2 F1F0E4E2 E3C5E2E3 4040005C E6870097  *..... ........OS10USTEST  ..W...*
000020 321F4040 40404040 40408480 C3C1E3C1 D3D6C74B E3C9C3C6 E4E2C5D9 40404040  *..          ..CATALOG.TICFUSER    *
000040 40404040 40404040 40404040 40404040 40404040 40404040 E4E2E3C5 E2E34BE4  *                          USTEST.U*
000060 D7E2E3D9 C5C1D44B C6C9D3C5 C4C1E3C1 40404040 40404040 40404040 40404040  *PSTREAM.FILEDATA                  *
000080 40404040 0000000A 5000001A 00210000 002A000E E4E2E3E6 D2F10190 00003030  *    ................USTWK1......*
0000A0 200E0000 00000000 01180000 00000000 00010000 001F0000 00030000 00000000  *................................*
0000C0 00000000 00030066 C8000000 00000000 00000000 00160000 00000000 00000000  *........H.......................*
0000E0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000  *................................*
000100 00000000 58000000 58000000 18150015 E4E2E3C6 C9D3C5C3 040F0000 00000008  *................USTFILEC........*
000120 0006E4E2 E3C5E2E3 4BE4D7E2 E3D9C5C1 D44BC6C9 D3C5C4C1 E3C14040 40404040  *..USTEST.UPSTREAM.FILEDATA      *
000140 40404040 40404040 40404040 40400004 96000400 00000000 00000000 00000000  *              ..................*
000160 00000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000  *................................*
      LINE 000180 SAME AS ABOVE
0001A0 00000000 00000000 00000000 00000000 00000000 00000000 00000000 0000      *..............................  *

IAM601   SMF RECORDS -- READ......8319  USED.........1  DROPPED.........0
IAM492   RECORD PRINT     FUNCTION ENDED   - 16.27.27 - CONDITION CODE 000

IAM499   IAMSMF(8.0/01P ) PROCESSING COMPLETED
```

Figure 26: Sample of PRINT Command Output

41.06 IAMSMF - QUERY COMMAND

QUERY COMMAND

The QUERY command is used to report on dataset activity as requested by dataset name or dataset group name from a history file (RECFM=VBS), an active SMF data recording file, or from a sequential file produced by the COPY command of the program itself. This command is quite useful for tracking down the jobs and/of TSO users that have been using a dataset, and also the jobs that defined and deleted the dataset.

QUERY	
[CHECKLENGTH]	**[,JOBNAMES=**jobname**]**
[,DSGROUPS=dsname]	**[,MAXCORE=**nnnn**]**
[,DSNAMES=dsname**]**	**[,MAXRECLENGTH=**nnnnn**]**
[,ERRORPRINT]	**[,MAXSTACK=**nnnnn**]**
[,FROMDATE=yyyyddd**]**	**[,PRTLENGTH=**nnnnn**]**
[,FROMDDNAME=ddname**]**	**[,RECSIZE(rrr)=**nnnnn**]**
[,GROUPNAMES=jobnames**]**	**[,TODATE=**yyyyddd**]**

Figure 27: IAMSMF QUERY Command Operands

QUERY COMMAND OPERANDS

Operand	Description
CHECKLENGTH	Specifies that SMF data records are to be validated against a table of minimum record lengths. If the user has modified the minimum record length or is executing in a non-compatible system, the correct record lengths may be specified by the RECSIZE operand. The default is that the length of an SMF record is not validated.
DSGROUPS=	Specifies that only records having a dataset name which begin with the given character string(s) will be processed. This operand specifies a partial dataset name from 1 to 44 characters in length. Up to 50 dataset groups may be specified for a single command if entered as follows: **DSGROUPS**=(dsname1,...,dsnamex) **NOTE:** If neither the DSGROUPS nor DSNAMES operand is specified, the QUERY command will be marked in error.
DSNAMES=	Specifies that only records having a dataset name which match the dataset name(s) specified will be processed. This operand specifies a complete dataset name from 1 to 44 characters in length. Up to 50 dataset names may be specified for a single command if entered as follows: **DSNAMES**=(dsname1,...,dsnamex) **NOTE:** If neither the DSGROUPS nor DSNAMES operand is specified, the QUERY command will be marked in error.
ERRORPRINT	Specifies that any SMF data record that causes an error during processing or fails length verification is to be printed. By default, records in error are not printed.

41.06 CONTINUED . . .

FROMDATE= Specifies the lower date limit of the SMF records that are to be selected, in the format of yyyyddd or yyddd. The shorter format assumes a prefix of 19.

The default, if the FROMDATE and/or TODATE operands are not specified, is that the date of the SMF record will not participate in the selection criteria.

FROMDDNAME= Specifies the DDNAME of the SMF file to be used as input to IAMSMF.

The default input DDNAME is SYSMF.

GROUPNAMES= Specifies that only records having a jobname which begin with the given character string(s) will be processed. This operand specifies a partial jobname from 1 to 8 characters in length. Up to 50 job groups and/or names may be specified for a single command if entered as follows:

GROUPNAMES=(jobname1,...,jobnamex)

The default, if neither the GROUPNAMES nor JOBNAMES operand is specified, is that the jobname will not participate in SMF record selection.

JOBNAMES= Specifies that only records having a jobname which match the jobname(s) specified will be processed. This operand specifies a complete jobname from 1 to 8 characters in length. Up to 50 jobnames may be specified for a single command if entered as follows:

JOBNAMES=(jobname1,...,jobnamex)

The default, if neither the GROUPNAMES nor JOBNAMES operand is specified, is that the jobname will not participate in SMF record selection.

MAXCORE= Specifies the maximum number of bytes of working storage available for various commands. The number may be any value from 1000 to 120000, inclusive.

The default is 2400 bytes.

MAXRECLENGTH= Specifies the largest SMF record that the program will process. The number may be any value from 16384 to 65536, inclusive.

The default is 16384 bytes.

MAXSTACK= Specifies the maximum number of compressed SMF records that can be retained in storage. The number can be any value from 100 to 50000, inclusive.

The default is 2500 records.

PRTLENGTH= Limits to the number of bytes specified to the amount of data to be printed if ERRORPRINT is indicated. The number may be any value from 32 to 65536, inclusive.

The default is 32768 bytes.

RECSIZE(rrr)= Specifies the minimum length of the SMF records type 'rrr' is to be set to the value 'nnnn'.

The default minimum record lengths for system generated SMF records are documented in IBM Systems Management Facilities manual.

TODATE= Specifies the upper date limit of the SMF records that are to be selected, in the format yyyyddd or yyddd. If the shorter form is used, the value is prefixed with a 19.

The default, if the FROMDATE and/or TODATE operands are not specified, is that the date of the SMF record will not participate in the selection criteria.

41.06 CONTINUED . . .

EXAMPLE A: QUERY

The example below provides sample JCL and control card input to find all of the activity against a group of datasets. Each SMF record encountered for a dataset in the specified group will be printed. This enables you to determine all of the jobs that actively used the dataset(s), based on the SMF data.

```
//IAMSMF   EXEC  PGM=IAMSMF,REGION=1M
//STEPLIB  DD    DISP=SHR,DSN=iam.load.lib
//SYSPRINT DD    SYSOUT=*
//SYSMF    DD    DISP=SHR,DSN=smf.data.set
//SYSIN    DD     *
       QUERY    DSG=IAMV
/*
```

Figure 28: Sample JCL for IAMSMF QUERY Command (EX4106A)

SAMPLE QUERY OUTPUT

```
IAM400 SMF REPORT/DATA EXTRACT PROGRAM-IAMSMF VER 8.0/01P INNOVATION DATA PROCESSING DATE 2002.218
IAM303 CARD IMAGE - *   QUERY DSG=IAMV                                   00120036*
IAM491 DATA SET QUERY   FUNCTION STARTED - 09.44.32

DSN---IAMV.KSD270.CLUSTER                DATE-------2002.218   TIME-----15.05.08
  JOB------KSD270      DDNAME--**ABSENT  DSORG----------AM     USE--------DEFINE

DSN---IAMV.KSD270.CLUSTER                DATE-------2002.218   TIME-----15.05.08
  JOB------KSD270      DDNAME--SYS00001  DSORG----------PS     USE--------OUTPUT
  EXCP COUNTS----      DISK-----------2

DSN---IAMV.KSD270.CLUSTER                DATE-------2002.218   TIME-----15.05.11
  JOB------KSD270      DDNAME--VSAMCRT1  DSORG----------PS     USE--------OUTPUT
  EXCP COUNTS----      DISK-----------1

DSN---IAMV.KSD270.CLUSTER                DATE-------2002.218   TIME-----15.05.13
  JOB------KSD270      DDNAME--VSAMCRT1  DSORG----------DA     USE--------OUTPUT
  EXCP COUNTS----      DISK----------43

DSN---IAMV.KSD270.CLUSTER                DATE-------2002.218   TIME-----15.05.22
  JOB------KSD270      DDNAME--**ABSENT  DSORG----------**     USE-------SCRATCH

IAM601   SMF RECORDS -- READ.....61228  USED.....51390  DROPPED........0
IAM492   DATA SET QUERY   FUNCTION ENDED   - 09.44.55 - CONDITION CODE 000

IAM499   IAMSMF(8.0/01P ) PROCESSING COMPLETED
```

EXAMPLE OF SELECTED FUNCTION QUERY

The QUERY command can also be used to find just selected functions for a dataset, by combining it with the COPY command. To find all the jobs that are defining the datasets that we want to convert, first copy just the VSAM DEFINE SMF records. Then run the QUERY command against the data in the SYSUT2 dataset. This results in a report consisting of only the jobs that defined the dataset(s).

```
//IAMSMF   EXEC  PGM=IAMSMF,REGION=1M
//STEPLIB  DD    DISP=SHR,DSN=iam.load.lib
//SYSPRINT DD    SYSOUT=*
//SYSMF    DD    DISP=SHR,DSN=smf.data.set
//SYSUT2   DD    UNIT=SYSDA,SPACE=(CYL,(10,5))
//SYSIN    DD    *
       COPY   RECTYPE=(61,63)
       QUERY  DSG=IAMV,FROMDD=SYSUT2
/*
```

Figure 29: Sample JCL to Find Jobs Defining VSAM Clusters (EX4106B)

41.06 CONTINUED . . .

SAMPLE QUERY OUTPUT

```
IAM400 SMF REPORT/DATA EXTRACT PROGRAM-IAMSMF VER 8.0/01P INNOVATION DATA PROCESSING DATE-2002.218
IAM303   CARD IMAGE - *   COPY  RECTYPE=(61,63)                                        *
IAM491   RECORD COPY      FUNCTION STARTED - 10.31.30

IAM601   SMF RECORDS -- READ.....63660  USED......3415  DROPPED.........0
IAM492   RECORD COPY      FUNCTION ENDED   - 10.31.52 - CONDITION CODE 000

IAM303   CARD IMAGE - *   QUERY DSG=IAMV,FROMDD=SYSUT2                                 *
IAM491   DATA SET QUERY   FUNCTION STARTED - 10.31.52

DSN---IAMV.KSD270.CLUSTER                DATE-------2002.218   TIME-----15.05.08
  JOB------KSD270     DDNAME--**ABSENT   DSORG----------AM     USE--------DEFINE

DSN---IAMV.KSD271.CLUSTER                DATE-------2002.218   TIME-----15.05.18
  JOB------KSD271     DDNAME--**ABSENT   DSORG----------AM     USE--------DEFINE

DSN---IAMV.KSD272.CLUSTER                DATE-------2002.218   TIME-----15.05.33
  JOB------KSD272     DDNAME--**ABSENT   DSORG----------AM     USE--------DEFINE

DSN---IAMV.KSD280.CLUSTER                DATE-------2002.218   TIME-----15.05.41
  JOB------KSD280     DDNAME--**ABSENT   DSORG----------AM     USE--------DEFINE

DSN---IAMV.KSD281.CLUSTER                DATE-------2002.218   TIME-----15.05.53
  JOB------KSD281     DDNAME--**ABSENT   DSORG----------AM     USE--------DEFINE


IAM601   SMF RECORDS -- READ......3415  USED......3415  DROPPED.........0
IAM492   DATA SET QUERY   FUNCTION ENDED   - 10.31.53 - CONDITION CODE 000

IAM499   IAMSMF(8.0/01P ) PROCESSING COMPLETED
```

Figure 30: Sample IAMSMF QUERY Report Output

41.07 IAMSMF - REPORT COMMAND

REPORT COMMAND

The REPORT command is used to print job step and/or TSO session related statistics, with optional dataset usage information. The reports can be used to obtain some basic performance information about selected jobs, including elapsed time, CPU time, and EXCP counts. This may be useful for comparing job performance between running with VSAM files versus running with IAM files. The reports can be requested by job or job group name from a history file (RECFM=VBS), an active SMF recording file, or from a sequential file produced by the COPY command of the program itself.

REPORT	
[ALLDSNAMES]	**[,MAXRECLENGTH=**nnnn **]**
[,CHECKLENGTH]	**[,MAXSTACK=**nnnn**]**
[,ERRORPRINT]	**[,NODSNAMES]**
[,FROMDATE=yyyyddd**]**	**[,PRTLENGTH=**nnnn**]**
[,FROMDDNAME=ddname **]**	**[,RECSIZE(rrr)=**nnnn **]**
[,GROUPNAME=jobname **]**	**[,TEMPDSNAMES]**
[,JOBNAMES=jobname **]**	**[,TODATE=**yyyyddd **]**

Figure 31: IAMSMF REPORT Command Operands

REPORT COMMAND OPERANDS

Operand	Description
ALLDSNAMES	Specifies that the IAMSMF report will show all the occurrences of any referenced dataset name. The EXCP count displayed will reflect each OPEN/CLOSE. The default is that dataset usage counts (EXCPs) are accumulated for each permanent dataset, and the dataset is only shown once. Temporary datasets are ignored. **NOTE:** This operand conflicts with the operand NODSNAMES.
CHECKLENGTH	Specifies that SMF data records are to be validated against a table of minimum record lengths. If the user has modified the minimum record length or is executing in a non-compatible system, the correct record lengths may be specified by the RECSIZE operand. The default is the length of an SMF record is not validated.
ERRORPRINT	Specifies that any SMF data record that causes an error during processing or fails length verification is to be printed. The default is records in error are not printed.
FROMDATE=	Specifies the lower date limit of the SMF records that are to be selected, in the form yyyyddd or yyyddd. The shorter form is prefixed with a 19. The default if the FROMDATE and/or TODATE operands are not specified, is that the date of the SMF record will not participate in the selection criteria.
FROMDDNAME=	Specifies the DDNAME of the SMF file to be used as input to IAMSMF. The default input DDNAME is SYSMF.

41.07 CONTINUED . . .

GROUPNAMES= Specifies that only those records having a jobname which begin with the specified character string(s) will be processed. This operand specifies a partial jobname from 1 to 8 characters in length. Up to 50 job groups and/or names may be specified for a single command if entered as follows:

GROUPNAMES=(jobname1,...,jobnamex)

The default, if neither the GROUPNAMES nor JOBNAMES operand is specified, is that the jobname will not participate in SMF record selection.

JOBNAMES= Specifies that only those records having a jobname which match the jobname(s) specified will be processed. This operand specifies a complete jobname from 1 to 8 characters in length. Up to 50 job names may be specified for a single command if entered as follows:

JOBNAMES=(jobname1,...,jobnamex)

The default, if neither the GROUPNAMES nor JOBNAMES operand is specified, is that the jobname will not participate in SMF record selection.

MAXRECLENGTH= Specifies the largest SMF record that the program will process. The number may be any value from 16384 to 65536, inclusive.

The default is 16384 bytes.

MAXSTACK= Specifies the maximum number of compressed SMF records that can be retained in storage. The number can be any value from 100 to 50000, inclusive.

The default is 2500 records.

NODSNAMES Specifies that the report will not show dataset oriented information.

By default, dataset usage information is displayed only once for each permanent dataset with accumulated statistics for multiple OPEN/CLOSE. Temporary datasets are ignored.

NOTE: This operand conflicts with the operands ALLDSNAMES and TEMPDSNAMES.

PRTLENGTH= Limits to the specified amount of data to be printed if ERRORPRINT is indicated. The number may be any value from 32 to 65536, inclusive.

The default is 32768 bytes.

RECSIZE(rrr)= Specifies the minimum length of the specified SMF record type is to be set to the value specified.

The default minimum record lengths for system generated SMF records are documented in the IBM Systems Management Facilities manual.

TEMPDSNAMES Specifies that reports produced will contain information on temporary as well as permanent datasets.

By default, temporary datasets are ignored.

NOTE: This operand conflicts with the operand NODSNAMES.

TODATE= Specifies the upper date limit of the SMF records that are to be selected, in the form yyyyddd or yyddd. The shorter form is prefixed with a 19.

The default, is that there is no upper date limit on the SMF records to be processed.

41.07 CONTINUED . . .

EXAMPLE A: IAMSMF REPORT COMMAND

The following JCL example demonstrates how to run the IAMSMF REPORT command. For this request, the GROUP keyword is specified with a value of KSD. This will cause IAMSMF to report on all jobs that have a job name beginning with KSD that are contained within the SMF data provided.

```
//IAMSMF   EXEC  PGM=IAMSMF,REGION=1M
//STEPLIB  DD    DISP=SHR,DSN=iam.load.lib
//SYSPRINT DD    SYSOUT=*
//SYSMF    DD    DISP=SHR,DSN=smf.data.set
//SYSIN    DD    *
 REPORT    GROUP=KSD
/*
```

Figure 32: Example of JCL to run IAMSMF REPORT Command (EX4107A)

SAMPLE IAMSMF REPORT OUTPUT

Shown below is a sample of what the REPORT output looks like, having been run with the above sample control card and JCL. Explanations of the data fields that appear on this report are described after the report example.

```
IAM400 SMF REPORT/DATA EXTRACT PROGRAM-IAMSMF VER 8.0/01P INNOVATION DATA PROCESSING DATE-2002.218
IAM303   CARD IMAGE - *   REPORT GROUP=KSD                                 00120036*
IAM491   SMF REPORT       FUNCTION STARTED - 11.07.54

JOB NAME.....KSD270   DATE.....2002.218
  STEP NAME..SUBSTEP1 PGM..IEFBR14  REGION REQ..960 REGION USE...504 COMP CODE..C-0000 PERFORM...1
       START.15.05.00 STOP.15.05.03 WALL.00.00.02.79  CPU.00.00.00.02 SRB.00.00.00.00
                                    TAT..00.00.00.11  TNA.00.00.02.68 TRT.00.00.00.11
       DD..JOBLIB     DSNAME...IAMX.MODTEST                           PO USE..INPUT    DISK..0
       SUMMARY OF I/O ACTIVITY.PAGE IN...0 PAGE OUT...0 TAPE........0 DISK.....0
                               SWAP IN...0 SWAP OUT...0 SWAP COUNT..0 SRVU.....259

  STEP NAME..KSD270B PGM..IAMTVSAM REGION REQ..960 REGION USE....844 COMP CODE..C-0000 PERFORM...1
       START..15.05.09 STOP.15.05.15 WALL.00.00.06.09 CPU.00.00.00.57 SRB.00.00.00.04
                                     TAT..00.00.04.48 TNA.00.00.01.61 TRT.00.00.04.48
       DD..VSAMCRT1   DSNAME...IAMV.KSD270.CLUSTER                    DA USE..OUTPUT   DISK..82
           JOBLIB              IAMX.MODTEST                           PO      INPUT    DISK..58
       SUMMARY OF I/O ACTIVITY.PAGE IN...0 PAGE OUT...0 TAPE........0 DISK....140
                               SWAP IN...0 SWAP OUT...0 SWAP COUNT..0 SRVU....12716
```

Figure 33: Sample of an IAMSMF REPORT Command Output

41.07 CONTINUED . . .

IAMSMF REPORT FIELD DESCRIPTIONS

Field Name	Description
Job Name	Provides the name of the JOB being reported on.
Date	The date on which the job ran.
Step Name	Further detail is broken down by job step. For each job step, this specifies the name of that step.
PGM	Indicates the name of the program being executed, from the EXEC PGM= card.
Region REQ	Shows the amount of region below the line requested by this job step.
Region Use	Shows the actual amount of virtual storage used by this job step.
Comp Code	Indicates the completion code for the job step.
Perform	Indicates the performance group that the job step ran in.
Start	Starting time of the job step.
Stop	Ending time of the job step.
Wall	Elapsed time of the job step.
CPU	Total TCB time used by the job step.
SRB	Total SRB time used by the job step.
TAT	Transaction Active Time from the SMF record. This is defined as the time that the transaction was swapped in plus the time that the transaction was swapped out but ready to run.
TNA	Transaction Not Active Time, which is calculated by subtracting the Transaction Active Time from the total elapsed time of the job step.
TRT	The Transaction Residence Time from the SMF record. This is defined as being the time that the transaction was swapped in.
DD	Specifies the DD NAME detail. A line is included for each permanent DISK or TAPE dataset.
DSNAME	The name of the dataset referenced on the indicated DD card.
DSORG	The dataset organization.
USE	How the dataset was used, e.g. as INPUT or OUTPUT.
DISK or TAPE	Indicates whether the dataset was on a DISK or TAPE device, followed by the EXCP count.
Page In	The number of pages of virtual storage that were transferred from the paging dataset into real storage.
Page Out	The number of pages of virtual storage that were transferred from real storage to a paging dataset.
Tape	The total number of TAPE EXCP's.
Disk	The total number of DISK EXCP's.
Swap In	The total number of pages swapped in.
Swap Out	The total number of pages swapped out.
Swap Count	The number of times that the job step was swapped out.
SRVU	The number of Service Units used by the job step.

42.01 IAMSIMVS - IAM'S SPACE SAVINGS ANALYSIS PROGRAM

OVERVIEW VSAM datasets usually account for a large portion of the DASD space allocated within any shop. IAM's advanced file structure takes 20% to 40% less disk space than an equivalent VSAM dataset. Using IAM's proprietary compression techniques an additional 20% to 50% savings can be realized. IAMSIMVS projects the DASD space savings and resultant cost reductions a VSAM to IAM conversion can provide for your installation. IAMSIMVS's reports give estimated savings for individual datasets and then summarize these savings by device type for the installation as a whole. IAMSIMVS can also be used to help plan a conversion to IAM by identifying datasets whose conversion would result in the greatest savings.

FUNCTIONAL SUMMARY IAMSIMVS selectively reads the records contained in a VSAM cluster and simulates the conversion of that cluster to an IAM dataset. The simulation results in two sets of figures. One showing the results if the file were loaded in IAM's standard (uncompressed) format and the other the results if the file were loaded in IAM's compressed file format. The simulated IAM file is assumed to have been defined using the same IDCAMS parameters (SPACE, RECORDSIZE, FREESPACE, etc.) that would have established the VSAM cluster being analyzed.

DATA COMPRESSION IAM's Data Compression technique results in a 20% to 50% reduction in the size of the individual records contained in a file. Compression however, is not accomplished at the expense of increasing the CPU time it would have taken to process the file with VSAM normally. File processing with IAM's Data Compression takes considerably less CPU time than other vendor's VSAM compression packages. In fact, IAM's CPU time is so much less than VSAM's, IAM with data compression typically takes less CPU time than normal VSAM processing without data compression. Compare this to the fact that other data compression packages must add their CPU time to compress and decompress on top of VSAM's processing time. IAM can automatically compress data records based on a qualifying default file size, or individually on request. IAM attempts to compress the data following the key in each record. The record must contain at least 10 bytes of data following the key to qualify for compression. If a compressed record is larger than the original, IAM leaves the record uncompressed.

DATA SAMPLING The most accurate way to simulate a conversion is of course to read and process every record in a file. Reading every record in a large file however can be relatively time consuming. IAMSIMVS attempts to achieve a balance between exacting accuracy and reasonable run time by using only a sample of the records in a VSAM cluster. IAMSIMVS by default, limits its selection of the control areas it will read records from to a percentage of the control areas in the cluster. This technique limits the records actually read to a relatively small percentage of the total records the file actually contains. IAMSIMVS in this way is able to obtain both a good distribution in its sample of records and maintain an optimum level of performance. The sample rate IAMSIMVS will use is based upon the allocated size of the VSAM cluster whose conversion is to be simulated and may be anywhere from 10% to 100% of the records in the VSAM cluster, as shown in the following table.

Cylinders	Sampling Rate
1 - 9	100%
10 - 49	50%
50 - 99	20%
100 - nnnn	10%

In some unique circumstances, sampling may not be appropriate. For example, if a file was created with 100,000 records and all but the last 5,000 have since been deleted, sampling 10% of the file's Control Areas may not give the most accurate picture of the data that this file usually contains (i.e., most control areas are presently empty).

AUTOMATIC RELEASE IAMSIMVS will show you exactly how much of their allocated space your VSAM clusters are using for the data they contain. The remaining space, usually the result of over allocation, is wasted. IAM releases unused space when a file is initially loaded, automatically when the file is defined with secondary allocation.

42.01 CONTINUED . . .

SIZE ESTIMATES

Care has been taken to give you the most accurate estimate possible. However, IAMSIMVS estimates on the size of a converted IAM file are still only estimates and may even vary by as much as + or - five percent from a true IAM file for the following reasons:

- The VSAM catalog record contains the results of a DEFINE not the exact parameters used for the DEFINE.
- The VSAM cluster may contain records which vary greatly in size.

IAMSIMVS size estimates could be off by more than five percent from a true IAM file for the following reasons:

- The VSAM file varies greatly in size from day to day. When the file is converted to IAM it may contain a different number of records.
- The VSAM file contains a large number of deleted records when compared to the number of records initially loaded or inserted. Since these records are no longer on the file, IAM cannot determine their average record sizes. In addition, the maximum number of records the file may have contained cannot be determined.
- The VSAM file uses a small number of tracks as compared to the allocated space. For example, if the VSAM file was allocated with 1,000 tracks but is using only 20 tracks. IAM does not know if the over allocation was accidental or in anticipation of file growth.

REPORT FORMATS

IAMSIMVS's DataSet Report displays for each cluster reported on:

- The number of tracks allocated for the VSAM cluster.
- The number of tracks actually used by the VSAM cluster.
- An estimated number of tracks an equivalent IAM file will occupy.
- An estimated number of tracks a compressed IAM file will occupy.
- An estimated percent of the IAM savings over the VSAM used space. (or optionally the allocated space).
- The average and largest record size encountered sampling the file.
- Various VSAM file attributes (total records, key length, etc.)

IAMSIMVS also produces a Summary Report by DASD device type on:

- The total number of datasets processed.
- The total number of VSAM tracks allocated.
- The total number of VSAM tracks in use.
- VSAM's percent of tracks used as compared to allocated.
- The total savings, in tracks, a conversion to IAM can return.
- The total savings, in tracks, IAM's Data Compression can return.
- The dollar value, of the savings which can be realized by converting to IAM (with and without Data Compression).

ADDITIONAL CONSIDERATIONS

IAMSIMVS must open and read records from the specified VSAM clusters. Under a security package (RACF, Top Secret, ACF2, etc.) you must have proper access authorization to select a cluster for conversion simulation. If the IAMSIMVS job does not have authorization to read the specified VSAM clusters it will fail. IAMSIMVS does not need or use IAM's VSAM Interface (VIF).

COMMAND SUMMARY

IAMSIMVS has the following commands:

REPORT – Set processing defaults

SELECT – Identify those datasets which are to be used for simulation.

42.02 IAMSIMVS - JCL REQUIREMENTS

The JCL statements required to execute IAMSIMVS are as follows:

EXEC STATEMENT Specifies the name of the IAM simulation program - IAMSIMVS. For optimum VSAM performance, the region size for IAMSIMVS should be at least 8192K. For example:

```
//SIMVS     EXEC  PGM=IAMSIMVS,REGION=8192K
```

DD STATEMENTS

DD Name	Description
STEPLIB or JOBLIB	Indicates the library containing the IAM load modules. This statement is optional if the IAM library is in the link list, as recommended.
SYSUDUMP	Specifies the ABEND dataset used if major errors are detected. Usually a SYSOUT dataset.
SYSPRINT	Specifies where the IAMSIMVS control statements, messages, and reports are to be printed. Usually a SYSOUT dataset.
SYSIN	Specifies the input control statement dataset. Usually a DD * dataset.

NOTE: The IAMSMFVS VSAM Usage Analysis program can optionally generate IAMSIMVS SELECT command statements corresponding to the VSAM cluster's named in it's Cluster Size Report. (See Section 40.02, IAMSMFVS JCL Requirements regarding the optional SYSPUNCH DD Statement). To have a Space Savings Report produced by IAMSIMVS using these generated SELECT statements simply assign the IAMSIMVS SYSIN DD statement to the dataset that was written to by the IAMSMFVS SYSPUNCH DD Statement.

42.03 IAMSIMVS - REPORT COMMAND

REPORT COMMAND

The IAMSIMVS REPORT command is used to override processing defaults.

REPORT	
[COSTPERMB=nn **]**	**[,MAXDSN=**nnnn **]**
[,COST3380=nn **]**	**[,TYPE=**cccccccc **]**
[,COST3390=nn **]**	**[,$SIGN=**xxxx **]**
[,ESTIMATE=ccccc **]**	

Figure 34: IAMSIMVS Report Command Operands

42.03 CONTINUED . . .

OPERANDS The following operands may be specified on the REPORT command.

Operand	Description
COSTPERMB= **COST3380=**	Specifies the cost per megabyte for a 3380. In the Summary Report, IAMSIMVS will generate an estimate of the cost savings that could result if the selected VSAM clusters were converted to IAM. The default is $3 per megabyte. This value was derived from the average cost for all 3380 models available from both IBM and the third party market, weighted to reflect controller and operating costs.
COST3390=	Specifies the cost per megabyte for a 3390. In the Summary Report, IAMSIMVS will generate an estimate of the cost savings that could result if the selected VSAM clusters were converted to IAM. The default is $3 per megabyte. This value was derived from the average cost for all 3390 models available from both IBM and the third party market, weighted to reflect controller and operating costs.
ESTIMATE=	Specifies the base from which IAM is to determine its space savings estimates. Valid values are: **ALLOC** – Savings estimates are to be based upon the number of tracks allocated for the VSAM cluster. IAM automatically releases any unused space within the allocation. **USED** – Savings estimates are to be based only upon the number of tracks actually used by the data in the VSAM cluster. The default is USED.
MAXDSN=	Specifies the maximum number of datasets IAMSIMVS is prepared to simulate conversions of in one report. **NOTE:** If the MAXDSN value is exceeded, IAM will split the datasets among multiple reports. The default is 1000 datasets.
TYPE=	Identifies the format of IAM file the simulated conversion is to be for. The following options are supported: **IAM** – Simulate conversion to standard format **COMPRESS** – Simulate conversion compressed format. **BOTH** – Simulate conversion to both IAM standard format and IAM compressed formats. The default is BOTH.
$SIGN=	Specifies the unit of currency to be used when displaying the value of the disk space saved in IAMSIMVS's summary report. Specify 1 to 4 characters. If you change the currency sign, you should also adjust the cost per MB value to reflect the currency change. The default is '$'.

42.04 IAMSIMVS - SELECT COMMAND

SELECT COMMAND The IAMSIMVS SELECT command is used to identify those files for which a conversion simulation is to be performed. Additionally the SELECT command can be used to specify IAMSIMVS's sample rate and IAM file definition options. One or more SELECT statements must be specified.

SELECT	
DSNAMES=dsname	**[,OVERFLOW=**nnnnnn**]**
[,DSTYPE= KSDS] ESDS] ALL]	**[,PE=**nnnn**]**
[,BLOCKSIZE=nnnnn**]**	**[,SAMPLE=**nnn**]**
[,BUFND=nnn**]**	**[,VARIABLE]**
[,INTEGRATED=nn**]**	

Figure 35: IAMSIMVS Select Command Operands

OPERANDS The following operands may be specified on the SELECT command.

Operand	Description
DSNAMES=	Identifies those datasets for which a conversion to IAM is to be simulated. Up to 50 dataset names may be specified on each SELECT if entered as follows: **DSN**=(dsn,dsn,...,dsn) There is no default value for DSNAMES and it must be specified. **NOTE:** Although only 50 dataset names may be identified on each SELECT, all entries are tabled and will appear on the same report (up to the MAXDSN=value).
DSTYPE=	Specifies the type of VSAM files to be processed. **KSDS** – Process VSAM KSDS files only. **ESDS** – Process VSAM ESDS files only. **ALL** – Process KSDS and ESDS files. The default is ALL.
BUFND=	Sets the number of VSAM data buffers to be used when accessing the specified dataset. To obtain optimum performance, a default value will be set to a number large enough for VSAM to read an entire control area at a time.
BLOCKSIZE=	Establishes the blocksize or blocking factor to be used when simulating conversion to IAM. By default, IAMSIMVS will base its calculations on the VSAM cluster Control Interval (CI) size. IAM rounds the CI size value up to develop a blocksize that will fit most efficiently on the track a multiple number of times. A minimum of quarter track blocking (i.e. four blocks on a track) is used.
INTEGRATED=	Establishes the amount of free space to be simulated in each block to accommodate record inserts. This keyword is comparable to the IDCAMS FREESPACE(CI% ...) option. The default is to use the VSAM Cluster's Control Interval Percentage Free value.

42.04 CONTINUED . . .

OVERFLOW= Specifies the simulated number of blocks to be set aside for overflow records.

The default overflow value is established by taking the VSAM Cluster's Control Area Percentage Free value (FREESPACE (... CA%)) and multiplying it by the estimated number of records in the cluster's primary allocation.

PE= Establishes the simulated number of blocks IAM is to set aside for adds to the end of the file.

The default is 3 blocks.

SAMPLE= Specifies the percentage of records to be read from the VSAM cluster when simulating. Any value from 10 to 100 may be used.

The default sampling rates are based upon the size of the VSAM cluster as follows:

Cylinders	Sampling Rate
1 - 9	100%
10 - 49	50%
50 - 99	20%
100 - nnnn	10%

VARIABLE Identifies the file as containing variable length records.

Under normal processing, a file is assumed to have variable length records if the average and maximum record lengths (RECORDSIZE) specified for the VSAM file are not equal.

If message IAM318 is presented during processing the file was assumed to contain fixed length records, but the record lengths were not all equal. Should this message appear, rerun the simulation specifying the VARIABLE keyword. The values established by IAMSIMVS (overflow, blocking, etc.) will be different for a file containing fixed length records and one containing variable length records.

NOTE: If you decide to convert this file to a Compatible Format IAM file, the average record length in the IDCAMS DEFINE RECORDSIZE parameter must be changed to a value which is one less than maximum record length. If this is not done the load of the IAM file will fail.

42.05 IAMSIMVS - SAMPLE REPORTS

IAMSIMVS supplies the user with two basic reports;

- The Conversion Simulation Cluster/DataSet Report
- The Savings Summary Reports.

The Conversion / Simulation Cluster / DataSet Report

```
IAM400 VSAM ALLOCATION ANALYSIS - IAMSIMVS VER 8.0/01P - INNOVATION DATA PROCESSING DATE-yyyy.ddd PAGE nn
                                 IAM VS. VSAM ALLOCATION SIMULATION
                       VSAM TRKS  IAM TRKS  % SAVINGS    TOTAL    AVERAGE   MAX  KEYLN CISIZ        CI% CISPL
DATA SET NAME         ALLOC USED  STD COMPR STD COMPR   RECORDS   LARGEST LRECL  RKP  BLKSZ FLAGS  CA% CASPL
SMALLER.VSAM.DATASET    451  346  270    90  22    74    40496        242   250   20  4096   2     10     0
                                                      SAMPLE=50%      250          0  4096 IMBED  10     0
MID.SIZED.VSAM.KSDS.   1126 1111  780   510  30    54    46118        502   679   15  4096         10     0
  DATASET                                             SAMPLE=20%      670          1  4096         10     0
LARGER.VSAM.DATASET    2964 2949 2220   915  25    69   157405        600   600   20  4096          0     0
                                                      SAMPLE=10%      600          1  4096 IMBED   0     0
VSAM.KSDS.DATASET.      466  176   90    60  49    66    1001        1090  2048    8  1024  4,5   20     0
  WITH.SPANNED.RECS                                                  2048          4  1024 IMBED  15     0

FLAGS:
1) UNABLE TO ESTIMATE STANDARD IAM FILE ALLOCATION.

2) FILE WAS ASSUMED TO BE FIXED BUT WAS FOUND TO CONTAIN VARIABLE LENGTH RECORDS.

3) FILE INELIGIBLE FOR COMPRESSION BECAUSE THE LENGTH OF THE DATA PAST THE KEY WAS NOT MORE THAN 10 BYTES.

4) FILE HAS A HIGH DELETION RATE WHICH MAY RESULT IN THE IAM ESTIMATES BEING UNDERSTATED BY MORE THAN 5%.

5) FILE CONTAINS SPANNED RECORDS, IF SAMPLING WAS DONE, RECORDS WERE READ FROM THE BEGINNING OF THE FILE ONLY.
```

Figure 36: The Conversion Simulation Cluster / DataSet Report

The Savings Summary Report

```
IAM VS. VSAM ALLOCATION SUMMARY
DEVICE  TOTAL        VSAM TRACKS               IAM STANDARD TRACKS $            IAM COMPRESSED TRACKS $
TYPE    DSNS     ALLOCATED   USED  %USED    ALLOCATED  SAVED %SAVED  SAVED    ALLOCATED  SAVED  %SAVED  SAVED
3380     4           5,007  4,582     92        3,360  1647      36   $264        1,575  3,432      75   $549

NOTE:  THE ESTIMATED SAVINGS ARE BASED UPON THE USED NUMBER OF VSAM TRACKS AT A RATE OF $4 PER MB.

THE ACTUAL TRACKS USED BY AN IAM FILE MAY VARY BY + OR - FIVE PERCENT FROM THE ESTIMATES SHOWN.
```

Figure 37: The Savings Summary Report

42.06 IAMSIMVS - EXAMPLES

EXAMPLE A: Simulate the conversion of a number of VSAM files to IAM. Use the default values for the simulation.

```
//SIMULATE   EXEC  PGM=IAMSIMVS
//STEPLIB     DD   DISP=SHR,DSN=iam.library <==USER CHANGE
//SYSPRINT    DD   SYSOUT=*
//SYSUDUMP    DD   SYSOUT=*
//SYSIN       DD   *
    SELECT DSN=(VSAM.KSDS.FILE1,VSAM.KSDS.FILE2,
    VSAM.KSDS.FILE3,VSAM.KSDS.FILEN)
    SELECT DSN=(PAYROLL.VSAM.FILE,CICS.VSAM.FILE)
    ...
    SELECT DSN=(MASTER.VSAM.FILE,TABLE.VSAM.FILE)
/*
```

Figure 38: Example 1 of running IAMSIMVS (EX4206A)

EXAMPLE B: Simulated the conversion of four large VSAM clusters. The default values will be used for the first two clusters. The third cluster if converted to IAM, will need a larger amount of overflow set aside for inserts. Use a sample rate of 100% instead of the default of 10% for the fourth file, which is known to have variable length records which vary greatly in size.

```
//SIMULATE EXEC PGM=IAMSIMVS
//STEPLIB   DD  DISP=SHR,DSN=iam.library <==USER CHANGE
//SYSPRINT  DD  SYSOUT=*
//SYSUDUMP  DD  SYSOUT=*
//SYSIN     DD  *
    SELECT DSN=(LARGE.VSAM.KSDS.FILE1,LARGE.VSAM.KSDS.FILE2)
    SELECT DSN=FILE.WITH.MANY.ADDITION,OVERFLOW=100000
    SELECT DSN=LARGE.FILE.WITH.STRANGE.DATA,SAMPLE=100
/*
```

Figure 39: Example 2 of running IAMSIMVS (EX4206B)

EXAMPLE C: Simulate the conversion of two large KSDS clusters. Since IAM normally releases the unused portion of its allocation, have IAMSIMVS estimate its savings based upon the number of the allocated VSAM tracks instead of the IAMSIMVS default value of tracks used. This will result in a maximum savings figure for the amount of disk space IAM can return. Using this savings figure allows a direct comparison of IAM's disk space savings to that provided by any other technique. In addition the cost of the DASD per million bytes (MB) is changed from $4 per MB to $6 per MB. Use the 10% default sampling rate and defaults for all other values.

```
//SIMULATE  EXEC PGM=IAMSIMVS
//STEPLIB    DD  DISP=SHR,DSN=iam.library <==USER CHANGE
//SYSPRINT   DD  SYSOUT=*
//SYSUDUMP   DD  SYSOUT=*
//SYSIN      DD  *
    REPORT ESTIMATE=ALLOC,COST=6
    SELECT DSN=(CICS.MASTER.FILE1,CICS.MASTER.FILE2)
/*
```

Figure 40: Example 3 of running IAMSIMVS (EX4206C)

42.06 CONTINUED . . .

EXAMPLE D: Use IAMSMFVS to produce a Cluster Size Report ranking all (up to 500) of the KSDS Clusters identified in the currently available SMF history file. Then use IAMSIMVS to produce a detail DataSet Report showing the amount of savings a conversion to IAM would produce for each of these clusters and a Summary Saving Report - showing the overall saving the site would receive if all of these clusters were converted to IAM. Since many of the clusters are over allocated and this space can be recovered by IAM, estimate IAM's savings based on tracks allocated instead of tracks used. IAM will RELEASE the unused portion of an allocation.

```
//*
//*        CREATE IAMSIMVS SELECT STATEMENTS
//*
//IAMSMFVS EXEC PGM=IAMSMFVS,REGION=2048K
//STEPLIB   DD  DISP=SHR,DSN=iam.library <==USER CHANGE
//SYSMF     DD  DISP=SHR,DSN=...         <==POINTS TO SMF DATA
//SYSPRINT  DD  SYSOUT=*
//SYSUDUMP  DD  SYSOUT=*
//SORTIN    DD  UNIT=SYSDA,SPACE=(CYL,(25,5))
//SORTOUT   DD  UNIT=SYSDA,SPACE=(CYL,(25,5))
//SORTLIB   DD  DISP=SHR,DSN=SYS1.SORTLIB
//SORTWK01  DD  UNIT=SYSDA,SPACE=(CYL,(25,5))
//SORTWK02  DD  UNIT=SYSDA,SPACE=(CYL,(25,5))
//SORTWK03  DD  UNIT=SYSDA,SPACE=(CYL,(25,5))
//SYSOUT    DD  SYSOUT=*
//SYSPUNCH  DD  DSN=VSAM.CLUSTER.NAMES,DISP=(,CATLG),
//              UNIT=SYSDA,SPACE=(CYL,(5,5))
//SYSIN     DD  *
    REPORT  DSORG=AM,MAXDSNS=6000,MAXREPORT=500
/*
//*
//*        REPORT THE SAVINGS AN IAM CONVERSION WILL PROVIDE
//*
//SAVINGS  EXEC PGM=IAMSIMVS
//STEPLIB  DD   DISP=SHR,DSN=iam.library <==USER CHANGE
//SYSPRINT DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//SYSIN    DD   *
    REPORT ESTIMATE=ALLOC
//         DD   DSN=VSAM.CLUSTER.NAMES,DISP=SHR  <==SEE NOTE1
//
```

Figure 41: Example 4 of running IAMSIMVS using IAMSMFVS to create control cards (EX4206D)

NOTE1: The first step executes IAMSMFVS to create a dataset containing formatted IAMSIMVS Select Command statements for the 500 largest VSAM clusters. The second step then reads this dataset to determine the names of the VSAM clusters to simulate a conversion for this dataset is concatenated after the instream REPORT Command Statement, which is used to establish default processing parameters.

45.01 IAMRECVR - RECOVERY PROGRAM OVERVIEW

OVERVIEW

To assist in recovery of valuable data, IAM includes a special function program, IAMRECVR, which can often create a backup copy of the data in an IAM file that is unreadable by normal means. The IAMRECVR program has been designed to aid users in the recovery of the contents of IAM files that have become unusable. Such problems can arise due to hardware or media failures, due to improper sharing of the dataset for update, or due to an application or system software error or failure.

IAMRECVR is a tool to assist in the recovery of IAM datasets. IAMRECVR is not a replacement for having established data recovery procedures, but rather is a program that can be used as a part of data recovery. It is quite important to have procedures in place to recover data should unforeseen problems occur, which include periodic backups of the dataset, and possibly transaction journals or logs of updates to the dataset.

IAMRECVR reads the IAM file using a high performance EXCP technique. Only blocks determined to match the IAM file specifications are processed. Any block determined to be unreadable due to physical damage to the device (e.g.: data check) or due to corruption of the record formatting within the block will be skipped. Appropriate error messages describing any of the errors encountered will be printed.

On a recovery process, if there are records in Independent or Extended Overflow, the sequential output file will need to be sorted. IAMRECVR can do that automatically, and will then report on any duplicate records encountered, which can optionally be written out to a log dataset. The sequential output dataset from IAMRECVR can subsequently be used to REPRO the data back into the recovered dataset.

Other capabilities offered by IAMRECVR include:

- Decompress and write out a file of uncompressed data records from a backup of an IAM file that was done with the BACKUPCOMPRESSED feature.
- Validate the integrity of the data blocks within an IAM dataset.
- Print out portions of the IAM dataset in a dump format.
- Create an image of the file structure of an IAM dataset without copying any data other than the key.
- Produce a report similar to a LISTCAT with the IAM file characteristics and statistics.

If you are preparing to do an IAM file recovery, be sure to read Section 10.87, Recovering IAM DataSets. That section provides an explanation, a technique, and several examples of using IAMRECVR to recover an IAM dataset.

COMMAND SUMMARY

The IAMRECVR program has the following commands / functions.

APPLY: – Copy records from a SPANOUT or LOG dataset created by the RECOVER command, into an IAM or VSAM dataset.

DIAGNOSE: – Validates the basic data integrity of an IAM file.

IAMSTRUCTURE: – Dump the structure of an IAM dataset. The only portion of the data retained in the backup will be the key.

LIST: – Display the attributes of an IAM dataset.

PRINT: – Print out selected portions or an entire IAM dataset, in a dump format.

RECOVER: – Reads a damaged IAM file, producing a sequential (or IAM / VSAM) dataset containing the records that IAMRECVR is able to read.

45.02 IAMRECVR - JCL REQUIREMENTS

The JCL statements required to execute IAMRECVR are as follows:

EXECUTE STATEMENT Specifies the name of the IAM recovery program — IAMRECVR. For a file recovery operation, sufficient storage on the REGION parameter must be specified to include storage for a SORT, which may need to be called if there are records in Extended or Independent Overflow. A typical execute statement would be:

```
//RECOVER  EXEC PGM=IAMRECVR,REGION=2048K
```

DD STATEMENTS The following table identifies the DD statements required for running IAMRECVR.

DD Name	Description
STEPLIB or JOBLIB	An optional DD statement that specifies the library containing the IAM program load modules. This DD is not necessary if IAM is in the Link List, as is recommended. If a STEPLIB or JOBLIB is used, then the IAM load library must be APF authorized.
SYSPRINT	Specifies where the IAMRECVR messages are to be printed. Usually a SYSOUT dataset.
DISKIN	Specifies the IAM dataset to be recovered, listed or printed.
DISKOUT	Specifies the new IAM file to be created from the damaged IAM file. The new IAM file will have the same characteristics as the IAM file being recovered.
TAPEOUT	Specifies the sequential recovery file to be created by IAMRECVR on either a tape or direct access device. If a direct access device is used, enough space must be allocated to contain a sequential copy of the IAM file. The use of secondary allocation values is permitted and encouraged.
VSAMOUT	Specifies the new VSAM file to be loaded from the damaged IAM file.
SPANOUT	Specifies a sequential dataset used to hold spanned records for recovery. This file can reside on either tape or direct access, and will be used by the APPLY command after reloading the recovered dataset containing the non-spanned records from the TAPEOUT file.
LOG	For the RECOVER option, specifies a sequential file where the duplicates, if any, are to be stored. For the APPLY option, specifies the sequential file to be used as input.
SYSIN	Specifies the input control statement dataset. Usually a DD * dataset.
SORTWKnn SORTLIB	These DD statements may be required if you are running the RECOVER command. Be sure to provide adequate SORT work space for the file that you are recovering. Refer to documentation of your sort for additional information.

45.03 IAMRECVR - APPLY COMMAND

APPLY COMMAND STATEMENT

The APPPLY command reads the LOG dataset created by the RECOVER command, and will either add or replace records in the IAM dataset from the LOG dataset. Apply is similar in function to an IDCAMS REPRO REPLACE. The APPLY command is used when a RECOVER command found duplicate records in the original IAM dataset to copy those records into the recovered dataset. The duplicate records will be copied from the LOG dataset that was created by the RECOVER command if DUPLICATES=LOG was specified.

For more information on how the APPLY command fits into the recovery process for an IAM dataset, refer to section 10.87 Recovering IAM DataSets.

APPLY	
[AUDIT=ccccc**]**	**[,IAMDDNAME=**ddname **]**
[,LOGDDNAME=ddname**]**	**[,OUTPUTFILE=**cccc **]**
[,PRTLENGTH=nnnnn **]**	**[SPANDDNAME=**ddname **]**
[,SPANNED]	**[,VSAMDDNAME=**ddname **]**

Figure 42: IAMRECVR APPLY Command Operands

45.03 CONTINUED . . .

APPLY COMMAND OPERANDS

Operand	Description
AUDIT=	Defines the audit trail requirements for records processed by the APPLY command. Possible values are: **KEY –** Print the key from the data record. **NONE –** No audit trail is produced. **RECORD** – Print the entire data record. The default is KEY.
IAMDDNAME=	Defines the DDNAME of the IAM file that is to be updated. The default is DISKOUT.
LOGDDNAME=	Defines the DD Name of the log file to be used as input to the APPLY operation. This file is from the LOG output of a RECOVER command. The default DD Name is LOG.
OUTPUTFILE=	Specifies the access method to use for the target dataset. Valid values are: **IAM –** Use the native mode IAM interface. Cannot be used for Enhanced format files, must specify VSAM. **VSAM –** This can be specified for either IAM or VSAM datasets. This option will cause IAMRECVR to use VSAM I/O macros.
PRTLENGTH=	Limit the amount of information printed in the AUDIT trail to this value or the length of the key or data, whichever is smaller. The default is 32768.
SPANDDNAME=	Specifies the DD name of the dataset containing the spanned records. This is the dataset that was created by the IAMRECVR RECOVER command. The default is SPANOUT.
SPANNED	Specifies that the APPLY command is to apply the spanned records contained in the dataset specified by the SPANOUT DD card. The LOG dataset will not be used in the APPLY SPANNED operation.
VSAMDDNAME=	Specifies the DD name of the IAM or VSAM dataset that is to be updated when OUTPUTFILE=VSAM is specified. The default is VSAMOUT.

45.03 CONTINUED . . .

EXAMPLE A: APPLY COMMAND

A basic example of using the APPLY command is shown below. For information on how the APPLY command should be used for a file recovery, refer to Section 10.87, Recovering an IAM DataSet.

```
//APPLYLOG  EXEC   PGM=IAMRECVR,REGION=4M
//SYSPRINT   DD    SYSOUT=*
//DISKOUT    DD    DISP=OLD,DSN=my.iam.new.dataset
//LOG        DD    DISP=SHR,DSN=my.iam.log.dataset
//SYSIN      DD    *
  APPLY
/*
```

Figure 43: Example of JCL to run the APPLY Command (EX4503A)

EXAMPLE B: APPLY SPANNED RECORDS

Shown below is an example of using the APPLY command to add the spanned records back into the recovered dataset. Prior to executing this APPLY command, the file must be reloaded with the non-spanned records from the TAPEOUT file. Subsequently, the following APPLY command can be used to put the spanned records back into the recovered dataset.

```
//APPLYSPN  EXEC   PGM=IAMRECVR,REGION=4M
//SYSPRINT   DD    SYSOUT=*
//DISKOUT    DD    DISP=OLD,DSN=my.iam.new.dataset
//SPANOUT    DD    DISP=SHR,DSN=my.iam.spanout.dataset
//SYSIN      DD    *
  APPLY       SPANNED
/*
```

Figure 44: Example of JCL to run the APPLY SPANNED Command (EX4503B)

45.04 IAMRECVR - DECOMPRESS COMMAND

RECOVERY OF DATA THAT IS COMPRESSED

The IAM file recovery program, IAMRECVR is able to read sequential files created with the BACKUPCOMPRESSED feature, and write out a sequential file with the data uncompressed. This may be useful for when an application program needs to read the sequential file from FDRREORG or IDCAMS REPRO of the IAM file. This is facilitated with the new command, DECOMPRESS.

DECOMPRESS COMMAND STATEMENT

The DECOMPRESS command allows a compressed backup of an IAM file to be decompressed in the event that IAM is not available. The IAM VSAM Interface (VIF) does not have to be active.

DECOMPRESS	
[FROMDDNAME= ddname **]**	**KEYLEN=** nnn
,RKP= nnnnn	**[,SCAN]**
[,TODDNAME= ddname **]**	

Figure 45: IAMRECVR DECOMPRESS Command Operands

DECOMPRESS COMMAND OPERANDS

The table below contains descriptions of the keywords for the DECOMPRESS command of IAMRECVR. The minimal abbreviation for each operand keyword is underlined.

Keyword	Description
FROMDDNAME=	Defines the DDNAME of the compressed IAM backup file. The default is DISKIN.
KEYLEN=	Required operand that specifies the length of the key within the data records in the IAM file. This value can be obtained from an IAMINFO or LISTCAT report on the original IAM dataset.
RKP=	Required operand that specifies the relative location of the key within the data record in the IAM file. This value can be obtained from an IAMIFNO or LISTCAT report on the original IAM dataset.
SCAN	Allows a compressed backup file to be read and all records decompressed, but an output file will not be created. SCAN can be used to verify a compressed backup file.
TODDNAME=	Defines the DDNAME of the output uncompressed flat file. The default is TAPEOUT.

EXAMPLE A: DECOMPRESS

Assume an IAM file with 100 byte records, an 8 byte key length with a relative key position (RKP) of 10 has been backed up by an IDCAMS REPRO with the BACKUPCOMPRESSED option, and is now needed in it's uncompressed state.

```
//DECOMPRS EXEC PGM=IAMRECVR
//SYSPRINT  DD  SYSOUT=*
//DISKIN    DD  DSN=my.seqfile,DISP=OLD
//TAPEOUT   DD  DSN=my.uncomp.seqfile,DISP=(,CATLG),
//     DCB=(RECFM=VB,LRECL=104,BLKSIZE=32760),UNIT=TAPE
//SYSIN     DD  *
      DECOMPRESS KEYLEN=8,RKP=10
/*
```

Figure 46: Example of JCL and Control Card for Decompress (EX4504A)

45.05 IAMRECVR - DIAGNOSE COMMAND

DIAGNOSE COMMAND

The DIAGNOSE command will read through the entire IAM file, and report on any errors that are encountered. The DIAGNOSE function reads the IAM dataset without using the IAM access method, and verifies the read integrity of each data block. DIAGNOSE can detect physical I/O errors, validate the format of all blocks containing user data, and verify that all records can be uncompressed.

The DIAGNOSE command is not able to detect problems with the control information saved for Independent Overflow, or the Extended Index. The DIAGNOSE will detect out of sequence records and duplicate records contained within any particular data block, however it is not able to detect for duplicate records that may exist in other areas of the file, such as in the Overflow areas.

DIAGNOSE	**[FROMDDNAME=** ddname **]**

Figure 47: IAMRECVR DIAGNOSE Command Operands

DIAGNOSE COMMAND OPERANDS

The following operand may be specified with the DIAGNOSE subcommand.

Operand	Description
FROMDDNAME=	Defines the DDNAME of the IAM file that is to be diagnosed. The default is DISKIN.

EXAMPLE A: DIAGNOSE

The following example demonstrates how to run an IAMRECVR DIAGNOSE command.

```
//DIAGNOSE EXEC  PGM=IAMRECVR
//SYSPRINT  DD   SYSOUT=*
//DISKIN    DD   DISP=SHR,DSN=my.iam.data.set
//SYSIN     DD   *
     DIAGNOSE
/*
```

Figure 48: Example of the IAMRECVR Diagnose Command (EX4505A)

45.06 IAMRECVR - IAMSTRUCTURE COMMAND

IAMSTRUCTURE COMMAND

The IAMSTRUCTURE command is used to copy all IAM KEY and INDEX information to a sequential output file. This output file can then be sent to Innovation to recreate as closely as possible the users file, without having to send any actual data. IAMSTRUCTURE will copy the entire KEY of each record, and the record length.

IAMSTRUCTURE will allow Innovation to recreate the logical structure of the IAM file to help with problem determination, and for testing of new enhancements to IAM. The output of the IAMSTRUCTURE command is a physical sequential (PS) file, which can be browsed under TSO to verify that no confidential information is being copied.

The IAMSTRUCTURE command has no operands. The input IAM file is specified on the DISKIN DD statement, and the sequential output file is specified on the TAPEOUT DD statement. An example of the JCL and control card to run IAMSTRUCTURE is shown below.

EXAMPLE A: IAMSTRUCTURE

```
//DUMPSTRC  EXEC PGM=IAMRECVR,REGION=4M
//SYSPRINT   DD  SYSOUT=*
//DISKIN     DD  DISP=SHR,DSN=my.iam.file
//TAPEOUT    DD  DISP=(,CATLG),DSN=my.iam.structre.file,
//               UNIT=SYSDA,SPACE=(CYL,(5,2))
//SYSIN      DD  *
IAMSTRUCTURE
/*
```

Figure 49: Sample JCL for the IAMSTRUCTURE Command (EX4506A)

45.07 LIST COMMAND

LIST COMMAND

The LIST command is used to list the characteristics of an IAM file. The format of the report is almost identical to the report on IAMPRINT from a LISTCAT ALL. The LISTCAT ALL will provide more information than the IAMRECVR LIST command, so use of a LISTCAT is recommended above the IAMRECVR LIST command.

LIST	**[FROMDDNAME=** ddname **]**

Figure 50: IAMRECVR LIST Command Operands

LIST COMMAND OPERANDS

The following operand may be specified with the LIST command.

Operand	Description
FROMDDNAME=	Defines the DDNAME of the IAM file that is to be listed. The default is DISKIN.

EXAMPLE A: LIST COMMAND

```
//LISTIAMF EXEC  PGM=IAMRECVR,REGION=1M
//SYSPRINT  DD   SYSOUT=*
//DISKIN    DD   DISP=SHR,DSN=my.iam.file
//SYSIN     DD   *
   LIST
/*
```

Figure 51: Example of JCL for LIST Command (EX4507A)

45.08 IAMRECVR - PRINT COMMAND

PRINT COMMAND The PRINT command is used to print various areas or blocks in dump format from an IAM file. A list of the file characteristics is produced upon completion.

PRINT	
[ALLBLKS]	**[,DATA]**
[,EXKEYS]	**[,EXTENDED]**
[,FBLK=nnnnnn**]**	**[,FROMDDNAME=**ddname**]**
[,IDPINQ]	**[,KEYS]**
[,MAXBLKS=nnnnn **]**	**[,OFLOW]**
[,PE]	**[,PRTLENGTH=**nnnnn **]**
[,TBLK=nnnnn **]**	

Figure 52: IAMRECVR PRINT Command Operands

45.08 CONTINUED . . .

PRINT COMMAND OPERANDS

Operand	Description
ALLBLKS	Specifies that all (or portions of all) blocks in the IAM file are to be printed. **NOTE:** Use of this operand conflicts with DATA, FBLK, KEYS, OFLOW, PE, and TBLK.
DATA	Specifies that all (or portions of all) prime data blocks in the IAM file are to be printed. **NOTE:** Use of this operand conflicts with ALLBLKS, FBLK, and TBLK.
EXKEYS	For Enhanced Format files, specifies that the PE index blocks and the Overflow RBN blocks are to be printed.
EXTENDED	For Enhanced Format files, specifies that all of the blocks (data and index) in the extended area of the file will be printed. These include Extended Overflow and PE blocks.
FBLK=	Specifies the From block number, relative to 1, from which printing is to begin. **NOTE:** Use of this operand conflicts with ALLBLKS, DATA, KEYS, OFLOW, and PE.
FROMDDNAME=	Defines the DDNAME of the IAM file that is to be listed. The default is DISKIN.
IDPINQ	Specifies the IAM control block is to be printed.
KEYS	Specifies that all (or portions of all) key blocks in the IAM file are to be printed. **NOTE:** Use of this operand conflicts with ALLBLKS, FBLK, and TBLK.
MAXBLKS=	Specifies the maximum number of blocks to be printed from each area selected in the IAM file. The default is the number of blocks in the area selected or the entire file if ALLBLKS is specified, excluding the IAM control block.
OFLOW	Specifies that all (or portions of all) Independent Overflow blocks in the IAM file are to be printed. **NOTE:** Use of this operand conflicts with 'ALLBLKS', 'FBLK', and 'TBLK'.
PE	Specifies that all (or portions of all) Prime Extension blocks in the IAM file are to be printed. **NOTE:** Use of this operand conflicts with 'ALLBLKS', 'FBLK', and 'TBLK'.
PRTLENGTH=	Limit the amount of data printed for each block to this value or the length of block, whichever is smaller. The default is 32768.
TBLK=	Specifies the block number, relative to 1, at which printing is to end. NOTE: Use of this operand conflicts with 'ALLBLKS', 'DATA', 'KEYS', 'MAXBLKS', 'OFLOW', and 'PE'.

45.08 CONTINUED . . .

EXAMPLE A: PRINT The example below demonstrates two different forms of the PRINT command. The first prints out the IAM control information blocks, in a dump format. The second prints out selected data blocks from the IAM dataset. The use of the PRINT command in problem diagnosis and recovery is discussed in Section 10.87, Recovering an IAM DataSet.

```
//PRINTIAM EXEC PGM=IAMRECVR,REGION=4M
//SYSPRINT  DD  SYSOUT=*
//DISKIN    DD  DISP=SHR,DSN=my.iam.dataset
//SYSIN     DD  *
      PRINT  IDPINQ
      PRINT  FBLK=100,MAXBLKS=10
/*
```

Figure 53: Example of the Print Command under IAMRECVR (EX4508A)

45.09 IAMRECVR - RECOVER COMMAND

RECOVER COMMAND

The RECOVER command is used to read an IAM dataset, which may be damaged or corrupted, and copy the records it is able to read into another dataset. Any errors encountered reading the input IAM dataset will be reported on, and may result in the loss of some data records if the errors cause some of the records to be unreadable. The output dataset can be a sequential dataset, an IAM dataset, a VSAM cluster, or a combination of sequential and IAM or VSAM. While there are several choices for the type of output dataset, Innovation recommends using only a sequential output dataset. The sequential dataset can then be copied into an IAM or VSAM dataset using IDCAMS REPRO.

The RECOVER command may not detect some of the errors, particularly if they occur within the overflow control information or extended index areas. Such errors may result in being unable to open the IAM dataset through normal programs. IAMRECVR does not rely on that information to open or read the dataset. The RECOVER will still work and be valid, even though no errors were detected within the data blocks.

If the input IAM dataset has records in Extended Overflow, or for Compatible format files in Independent Overflow, the output file will have to be sorted. The RECOVER command can automatically invoke the SORT, and it is highly recommended that the SORT be done by IAMRECVR. You will need to provide IAMRECVR with sufficient SORT work space based on the size of the file that is being recovered.

For some good examples of procedures to follow when recovering IAM datasets, be sure to read Section 10.87 of the manual, Recovering IAM DataSets. A full explanation of how to use the RECOVER command is provided there, along with several examples.

RECOVER

[AUDIT=cccccc **]**	**[,BLKSIZE=**nnnnn **]**
[,COMPRESSED]	**[,DUPLICATES=**cccccc **]**
[,FROMDDNAME=ddname **]**	**[,IAMDDNAME=**ddname **]**
[,KEYLEN=nnn **]**	**[,LOGDDNAME=**ddname **]**
[,LRECL=nnnnn **]**	**[,MAXBLKS=**n...n **]**
[,OUTPUTFILES=cccc **]**	**[,OVERFLOW=**nnnnn **]**
[,PRTLENGTH=nnnnn **]**	**[,RKP=**nnnn **]**
[,SORT=ccccc **]**	**[,SORTCORE=**n...n **]**
[,SORTMSG=cc **]**	**[,SORTPFX=**cccc **]**
[,SPANDDNAME=ddname **]**	**[,TODDNAME=**ddname **]**
[,VARIABLE]	**[,VSAMDDNAME=**ddname **]**

Figure 54: IAMRECVR Recover Command Operands

45.09 CONTINUED . . .

RECOVER COMMAND OPERANDS

Operand	Description
AUDIT=	Defines the audit trail requirements for duplicate records processed by the RECOVER command if SORT=IFREQ or SORT=YES and DUPLICATE= APPLY or if DUPLICATE= PRINT or DUPLICATE=LOG are specified. **KEY** – Print the key from the data record. **NONE** – No audit trail is produced. **RECORD** – Print the entire data record. The default is KEY.
BLKSIZE=	Specifies the actual blocksize of the IAM file. This value can be obtained from the run time statistics or a LIST command. **NOTE:** This value is ignored unless the IAM control record is destroyed, at which time it is required.
COMPRESSED	Identifies the file as containing compressed records. NOTE: This value is ignored unless the IAM control record is destroyed. This value is optional and is only used to request that the new IAM file is to have a compressed data structure. IAMRECVR can always detect a compressed record and decompress it.
DUPLICATES=	Defines the processing requirements for any duplicate records processed by the RECOVER command if SORT=IFREQ or SORT=YES is specified. **APPLY** – Update the IAM file being created with the duplicate records. Ignored unless 'OUTPUTFILES=IAM' or 'OUTPUTFILES=BOTH' is specified. **IGNORE** – Ignore duplicate records. **LOG** – Create a log dataset of any duplicate records. This dataset may later be used as input the 'APPLY' command. **PRINT** – Print any duplicate records. The default is PRINT.
FROMDDNAME=	Defines the DDNAME of the IAM file that is to be recovered. The default is DISKIN.
IAMDDNAME=	Defines the DDNAME of the IAM file that is to be created when OUTPUTFILES=IAM or BOTH is specified. The default is DISKOUT. **NOTE:** For Compatible format files, IAMRECVR will use the native IAM interface. For Enhanced format files, IAMRECVR uses the IAM VIF interface, so the file will have to be defined prior to running the RECOVER.
KEYLEN=	Specifies the length of the key within the data records in the IAM file. This value can be obtained from the run time statistics or a LIST command. **NOTE:** This value is ignored unless the IAM control record is destroyed, at which time it is required.

45.09 CONTINUED . . .

LOGDDNAME= Defines the DDNAME of the log file to be created.

The default is LOG.

LRECL= Specifies the logical record length of the data records in the IAM file. This value can be obtained from the run time statistics or a LIST command.

NOTE: This value is ignored unless the IAM control record is destroyed, at which time it is required.

MAXBLKS= Specifies the number of data blocks in the IAM file. This value can be obtained from the run time statistics or a 'LIST' command.

NOTE: This value is ignored unless the IAM control record is destroyed, at which time it is required.

OUTPUTFILES= Defines the output requirements for the RECOVER subcommand. Valid values are:

BOTH – Create both an IAM file and a sequential copy.

BOTHV – Create both an IAM file (using the VSAM interface) or a VSAM file, and a sequential copy.

IAM – Create only an IAM file.

SEQ – Create only a sequential copy of the recoverable data remaining in the IAM file.

VSAM – Create an IAM file (using the VSAM interface) or a VSAM file.

The default is SEQ.

OVERFLOW= Specifies the number of Independent Overflow blocks in the IAM file. This value can be obtained from the run time statistics or a LIST command.

NOTE: This value is ignored unless the IAM control record is destroyed, at which time it is required.

PRTLENGTH= Limit the amount of data printed for each block to this value or the length of block, which ever is smaller.

The default is 32768.

RKP= Specifies the relative location of the key within a data record in the IAM file. This value can be obtained from the run time statistics or a LIST command.

NOTE: This value is ignored unless the IAM control record is destroyed, at which time it is required.

SORT= Defines the output sorting requirements for the RECOVER command. Valid values are:

IFREQ – Sort the records only if sequence checks are encountered in the file.

NO – Do not sort the records.

YES – Sort the records.

The default is NO.

SORTCORE= Specifies the amount of storage the program SORT is to use if external sorting is required. The number maybe from 10000 to 8000000 inclusive.

The default is 100000.

45.09 CONTINUED . . .

SORTMSG= Specifies the message option to be used by the program SORT if external sorting is required.

AC – All messages to the console

AP – All messages to the printer (SYSOUT)

CC – Critical messages to the console

CP – Critical messages to the printer

NO – No messages to be produced

PC – Critical messages to both console and printer

The default is CC.

SORTPFX= Specifies the DDNAME prefix to be used by the program SORT if external sorting is required. If the string specified is less than 4 characters, a dollar sign ($) fill character will be used.

The default is SORT.

SPANDDNAME= Defines the DDNAME of the sequential output dataset created during recovery that contains the spanned records.

The default is SPANOUT.

TODDNAME= Defines the DDNAME of the sequential output dataset created during recovery.

The default is TAPEOUT.

VARIABLE Identifies the IAM file as having variable length records.

NOTE: This value is ignored unless the IAM control record is destroyed, at which time it is required.

VSAMDDNAME= Defines the DDNAME of the IAM or VSAM file to be created when OUTPUTFILES=VSAM or BOTHV is specified.

The default is VSAMOUT.

45.09 CONTINUED . . .

EXAMPLE A: RECOVER

The example below shows a basic RECOVER operation to a sequential dataset. Subsequent to the RECOVER, the sequential data would be copied into an IAM dataset using an IDCAMS REPRO. While it is rare that there will be duplicate records, they do occur on occasion. To save time when such a circumstance occurs, the example below includes logging the duplicate records to a LOG dataset. Such a situation does not necessarily indicate a problem with the dataset. If a record had to be moved from the block it was in into an overflow block, the overflow block is always immediately rewritten out to the dataset. At a subsequent point in time, the original data block is rewritten with the record deleted. So, there is an opportunity for a record with the same key to be duplicated in the dataset. IAM is able to handle this circumstance, and return the proper record. If there are duplicates, refer to Section 10.87, Recovering IAM DataSets for complete instructions and examples of recovering files from that situation.

```
//RECOVER   EXEC PGM=IAMRECVR,REGION=4M
//SYSPRINT   DD  SYSOUT=*
//SYSOUT     DD  SYSOUT-*
//DISKIN     DD  DISP=OLD,DSN=my.iam.cluster
//TAPEOUT    DD  DSN=my.seq.dataset,DISP=(,CATLG),
//               UNIT=SYSDA,SPACE=(CYL,(20,10))
//LOG        DD  DSN=my.duprec.dataset,DISP=(,CATLG),
//               UNIT=SYSDA,SPACE=(CYL,(2,1))
//SORTWK01   DD  UNIT=SYSDA,SPACE=(CYL,(20,10))
//SORTWK02   DD  UNIT=SYSDA,SPACE=(CYL,(20,10))
//SORTWK03   DD  UNIT=SYSDA,SPACE=(CYL,(20,10))
//$ORTPARM   DD  *                          ← Use for SyncSort
         EQUALS
/*
//DFSPARM    DD  *                          ← Use for DFSORT
         EQUALS
/*
//SYSIN      DD  *
   RECOVER       DUP=LOG
/*
```

Figure 55: Example of Running a Recover (EX4509A)

THIS PAGE WAS INTENTIONALLY LEFT BLANK

46.01 IAMISPF OVERVIEW

IAM includes an optionally installable set of ISPF panels that can be used to perform many utility functions against IAM and most VSAM datasets. The IAM ISPF interface offers an interactive, fill in the blanks type of functionality for various utility functions. The panels also include a complete ISPF tutorial that describes how to use the panels, as well as a complete description of the key IAM OVERFLOW statistics and OVERRIDE parameters.

DATASET NAMING CONVENTIONS

The following rules will apply in all of the IAM ISPF panels for specifying a dataset name. To specify a fully qualified dataset name, follow the TSO rules, which are to specify the dataset name within apostrophes. If no apostrophes are provided, then the dataset name will be automatically prefixed with the current TSO setting for dataset prefixing, which is normally the TSO userid. If the provided dataset name already has the current TSO prefix as the high level qualifier, then the prefix will not be added to the name.

FEATURES

The key features included in the IAM ISPF panels are:

- Definition of IAM datasets with full Override support.
- Definition of IAM Alternate Indexes and Paths
- Definition of VSAM clusters.
- Definition of VSAM Alternate Indexes and Paths
- Definition using an IAM or VSAM model dataset.
- Multi-Volume dataset support.
- SMS support.
- Building IAM or VSAM Alternate Indexes
- Deletion of Datasets, Clusters, Paths, and Alternate Indexes.
- Renaming of Datasets, Clusters, Paths, and Alternate Indexes.
- Copy/Move support of IAM datasets, VSAM ESDS's and VSAM KSDS's.
- IAM or VSAM datasets can be copied into or from sequential datasets.
- Full IAM dataset information.
- VSAM cluster information.
- Interactive execution of selected IAM utility functions.

Users that are familiar with ISPF and that have knowledge of IAM or VSAM should find the IAM ISPF panels easy to use. To illustrate some of the functions of the IAM ISPF panels, a demonstration is shown below.

46.02 IAMISPF PANELS

ISPF/PDF PRIMARY OPTION MENU

Shown below is an example of how the ISPF/PDF Primary Option Menu can be customized to include a selection for the IAM ISPF Dialog. By selecting option **I**, the user will be presented with the IAM Primary Option Menu.

```
Menu  Utilities  Compilers  Options  Status  Help
 ------------------------------------------------------------------------------
                            ISPF Primary Option Menu
 Option ===>

 0  Settings      Terminal and user parameters            User ID . : RAM2
 1  View          Display source data or listings         Time. . . : 10:59
 2  Edit          Create or change source data            Terminal. : 3278
 3  Utilities     Perform utility functions               Screen. . : 1
 4  Foreground    Interactive language processing         Language. : ENGLISH
 5  Batch         Submit job for language processing      Appl ID . : ISR
 6  Command       Enter TSO or Workstation commands       TSO logon : ISPFPROC
 7  Dialog Test   Perform dialog testing                  TSO prefix: RAM2
 8  LM Facility   Library administrator functions         System ID : CPUD
 9  IBM Products  IBM program development products        MVS acct. : MORSE.RI
 10 SCLM          SW Configuration Library Manager        Release . : ISPF 4.4
 A  FDR/ABR       FDR/ABR DASD MANAGEMENT
 I  IAM         - IAM utility functions
 U  UPSTREAM      FDR/UPSTREAM PC Backup/Recovery
```

Figure 56: Example of ISPF Primary Option Menu with IAM Option

IAM PRIMARY OPTION MENU

Shown below is the IAM primary option menu. The user specifies the desired function. For most functions, the user must also provide a dataset name, and other fields as appropriate.

```
-------------------------- IAM PRIMARY OPTION MENU --------------------------
OPTION ===>

   I      - Allocate (DEFINE) a new IAM Dataset                    Ver 8.01P
   V      - Allocate (DEFINE) a new VSAM Cluster
   D      - Delete a Dataset, Cluster, Path, or Alternate Index
   C      - Copy an IAM Dataset, VSAM KSDS, or VSAM ESDS
   M      - Move an IAM Dataset, VSAM KSDS, or VSAM ESDS
   R      - Rename a Dataset, Cluster, Path, or Alternate Index
   U      - Invoke an IAM Utility program
   blank  - Display IAM Dataset or VSAM Cluster Information


Enter dataset name (required for all options except U)
 Dataset Name       ===>
 Dataset Type       ===> C  C=Cluster X=AIX P=Path (options I, V Only)

Enter Model or New dsname (optional for options I and V, required for option R)
 Model|Newname      ===>


 Delete Confirmation ===> YES  Yes|No
```

Figure 57: IAM Primary Option Table

46.02 CONTINUED . . .

DEFINING AN IAM FILE

To define an IAM dataset perform the following steps:

- Type an **I** in the Option field of the IAM primary menu
- Fill in the dataset name field
- Type a **C** for the dataset type for an IAM ESDS or KSDS type of file
- Optionally, fill in the model dataset name field of an IAM or VSAM KSDS or ESDS
- Then press the ENTER key.

An example of a completed panel is shown below:

```
--------------------------- IAM PRIMARY OPTION MENU ---------------------------
OPTION ===> i

   I      - Allocate (DEFINE) a new IAM Dataset                    Ver 8.0/01P
   V      - Allocate (DEFINE) a new VSAM Cluster
   D      - Delete a Dataset, Cluster, Path, or Alternate Index
   C      - Copy an IAM Dataset, VSAM KSDS, or VSAM ESDS
   M      - Move an IAM Dataset, VSAM KSDS, or VSAM ESDS
   R      - Rename a Dataset, Cluster, Path, or Alternate Index
   U      - Invoke an IAM Utility program
   blank - Display IAM Dataset or VSAM Cluster Information


Enter dataset name (required for all options except U)
 Dataset Name       ===> myiam.cluster
 Dataset Type       ===> C  C=Cluster X=AIX P=Path (options I, V Only)

Enter Model or New dsname (optional for options I and V, required for option R)
 Model|Newname      ===>


Delete Confirmation ===> YES  Yes|No
```

Figure 58: Primary Option Panel set for Define

46.02 CONTINUED . . .

IAM DEFINITION PANEL

Shown below is the IAM dataset definition panel that comes up when defining an IAM Cluster. This panel supports both SMS managed and non-SMS managed allocations. Additionally, most IAM overrides can be specified by providing the appropriate values in the fields on the right hand side of the panel. By specifying YES in the Multi-Volume Allocation field, an additional panel will be displayed where additional volumes can be entered. Full information and examples of defining IAM datasets are presented in Section 10.20, Defining IAM DataSets.

```
------------------------------ DEFINE AN IAM FILE -----------------------------
 COMMAND ===>

 DataSet Name: RAM2.MYIAM.CLUSTER
                                       Multi-Volume Allocation ===> NO
 ALLOCATION
  Volume             ===> junk01   IAM OVERRIDES
  SMS Storage Class ===>              ANYVOL Unit        ===>
  SMS Data Class    ===>              Blocking Factor    ===>          1-15,>300
  SMS Mgmt Class    ===>              Overflow Records   ===>          0-2000000
  Cyls|Recs|Trks    ===> cyls         Var. Overflow      ===> YES      Yes|No
  Primary Space     ===> 2            Prime Extension    ===>          0-32767
  Secondary Space   ===> 1            Space Release      ===>          Yes|No
  Recatalog         ===> NO           Data Compress      ===>          Yes|No
 ATTRIBUTES                           Enhanced Format    ===> YES      Yes|No
  KSDS|ESDS         ===> ksds         Minbufno           ===>          1-32
  Max Recordsize    ===> 128          Maxbufno           ===>          1-32
  Avg Recordsize    ===> 100          PSEUDORBA (ESDS)   ===>          Yes|No
  Key Length        ===> 4            Extended ESDS      ===>          Yes|No
  Key Offset        ===> 8
  CI Size           ===> 4096        RETENTION
  CI/CA Free %      ===> 10  / 5      DAYS               ===>          0-9999
  Shareoption       ===> 2            EXPIRATION DATE    ===>          YYYY.DDD
```

Figure 59: Example of the IAM Define Panel

After filling in the necessary information with the desired values, press enter. IAM will return to the IAM Primary Option Menu, and indicate the status of the define request in the upper right corner. To display the file characteristics, make sure that the desired dataset name is filled in, then press enter.

IAM FILE CHARACTERISTICS PANEL

Shown below is an example of the IAM File Characteristics panel. This display is of the newly defined IAM dataset that has not been loaded with any data.

```
-------------------------- IAM FILE CHARACTERISTICS --------------------------
 COMMAND ===>
 Dataset Name: RAM2.MYIAM.CLUSTER

 Definition                         Allocation
    Record Length:        128          Volume:            JUNK01
    Record Format:   VARIABLE          Device Type:         3380
    Key Length:             4          Tracks in use:          0
    Key Offset:             8          Block Size:          4096
    Dataset Type:        KSDS          Blocking Factor:     4096
    Share Option:           2          Alloc Type:     CYLINDERS
    Release:              YES          Primary Alloc:          2
    Storage Class:                     Secondary Alloc:        1
    Data Class:                        Compressed Keys:       NO
    Mgmt Class:                        Compressed Data:       NO

 Statistics                         Extended Area
    Creation:        2002.321          Overflow Records:         0
    Expiration:      0000.000          Blocks Allocated:         0
    Last Reference:  2002.321          Blocks Used:              0
    Records:                0          Overflow Blocks:          0
    Deletes:                0          PE Blocks:                0
    Inserts:                0          Blocks Available:         0
    Updates:                0          Variable Overflow:      YES
    Minimum Buffers:
    Maximum Buffers:
    Extended File Fmt:    YES

 Associations
```

Figure 60: Sample IAM File Characteristics Panel for Newly Defined File

46.02 CONTINUED . . .

COPYING DATA INTO AN IAM FILE

Now that an IAM dataset has been defined, data can be loaded into the new dataset. In this example, data will be copied from a pre-existing sequential dataset into the new IAM file. Start at the IAM Primary Options panel, and do the following:

- Type a **C** for copy at the OPTION field.
- Type in the input dataset name in the dataset name field.
- Press enter.

```
--------------------------- IAM PRIMARY OPTION MENU ---------------------------
 OPTION ===> c

    I      - Allocate (DEFINE) a new IAM Dataset                    Ver 8.0/01P
    V      - Allocate (DEFINE) a new VSAM Cluster
    D      - Delete a Dataset, Cluster, Path, or Alternate Index
    C      - Copy an IAM Dataset, VSAM KSDS, or VSAM ESDS
    M      - Move an IAM Dataset, VSAM KSDS, or VSAM ESDS
    R      - Rename a Dataset, Cluster, Path, or Alternate Index
    U      - Invoke an IAM Utility program
    blank - Display IAM Dataset or VSAM Cluster Information


 Enter dataset name (required for all options except U)
  Dataset Name       ===> myiam.seq
  Dataset Type       ===> C  C=Cluster X=AIX P=Path (options I, V Only)

 Enter Model or New dsname (optional for options I and V, required for option R)
  Model|Newname      ===>


 Delete Confirmation ===> YES  Yes|No
```

Figure 61: Primary Option Panel set for Copy

When you press enter, IAM will display the following when the input dataset is a sequential file:

```
---------------- COPY AN IAM DATASET, VSAM ESDS, or VSAM KSDS -----------------
 COMMAND ===>

 Dataset Name ===> RAM2.MYIAM.SEQ


 The specified SOURCE  dataset is a NON-VSAM file. Would you like to continue
 with this REPRO? ===> Y (Y/N)
```

Figure 62: Sequential Input DataSet Confirmation Panel

To proceed with the copy operation, make sure that **Y** is specified, then press enter.

46.02 CONTINUED . . .

SETTING TARGET DATASET The next panel displayed is the COPY panel, on which the output (or target) dataset name is specified. Some of the other features, that will not be demonstrated here, include the Extended copy functions, where one can specify the number of records or key ranges to be copied. The other function is to drive from within the copy function a define of the target dataset. As the target dataset has already been defined, just enter the target dataset name, then press enter.

```
--------------- COPY AN IAM DATASET, VSAM ESDS, OR VSAM KSDS ----------------
 COMMAND ===>

 Source Non-VSAM File Name: RAM2.MYIAM.SEQ


 Enter target dataset or cluster name
   Target Dataset Name ===> myiam.cluster
   Extended Copy       ===> NO     (Yes|No    - To define copy selection criteria)


 Define Target Dataset ===> NO     (Yes|No    - If target does not exist)
 Target Dataset Type   ===>        (IAM|VSAM - If target is to be defined)
```

Figure 63: Copy Function Specification of Target DataSet

If the target dataset already has data in it, then IAM ISPF will display the following panel to confirm that the dataset should be overwritten:

```
------------------- TARGET DATASET OVERWRITE CONFIRMATION -------------------
 OPTION ===>

              The target dataset you have selected is not empty !!!

 DATA SET NAME:    RAM2.MYIAM.CLUSTER
 ACCESS METHOD:    IAM
 DATASET TYPE:     KSDS
 VOLUME:           JUNK01
 CREATION DATE:    2002.322

 OVERWRITE DATA SET ===> yes    (YES or NO)

    ENTER YES to overwrite the dataset.

    Enter NO or END command to cancel the COPY request.
```

Figure 64: Copy Overwrite Confirmation Panel

After the copy is completed, the IAM primary options panel is redisplayed, with a message in the upper right corner indicating that the copy is completed. If 10,000 or more records are being copied, there will be a status panel displayed indicating the progress of the copy. The status panel is updated after every 10,000 records have been copied.

46.02 CONTINUED . . .

IAM AIX DEFINE PANEL

To define an IAM Alternate Index dataset, start at the IAM Primary Options Panel.

- Type the option **I** in the option field, to define an IAM dataset.
- Specify the name of the alternate index in the dataset name field.
- Type **X** in the dataset type file, for an alternate index.
- Optionally fill in the name of a model alternate index.
- Then press Enter.

Shown below is the Define an IAM Alternate Index panel, which is shown after performing the above directions. Just below the alternate index dataset name, you must enter the name of the base IAM dataset that this dataset is an alternate index too. Then, complete the rest of the information including space allocation, attributes, any desired IAM overrides, and the retention period. Be sure to specify the alternate index attributes of whether or not this is an upgradeable alternate index, and whether or not the alternate index keys are unique. These fields are on the right side of the panel, below the override section.

```
----------------------- DEFINE AN IAM ALTERNATE INDEX -----------------------
 COMMAND ===>

 IAM Alternate Index DSN: RAM2.MYIAM.AIX
 Related IAM Cluster DSN: myiam.cluster
                                      Multi-Volume Allocation ===> NO
 ALLOCATION
  Volume            ===> JUNK01        IAM OVERRIDES
  SMS Data Class    ===>                ANYVOL Unit       ===>
  Cyls|Recs|Trks    ===> CYLINDERS      Blocking Factor   ===>          1-15,>300
  Primary Space     ===> 2              Var. Overflow     ===> YES      Yes|No
  Secondary Space   ===> 1              Space Release     ===>          Yes|No
  Recatalog         ===> NO             Data Compress     ===>          Yes|No
                                        Minbufno          ===>          1-255
 ATTRIBUTES                             Maxbufno          ===>          1-255
  Max Recordsize    ===> 73
  Avg Recordsize    ===> 13             Upgrade           ===> yes      Yes|No
  Key Length        ===> 4              Unique Keys       ===> no       Yes|No
  Key Offset        ===> 16
  CI Size           ===> 4096          RETENTION
  CI/CA Free %      ===> 10  / 5        DAYS              ===>          0-9999
  Shareoption       ===> 2              EXPIRATION DATE   ===>          YYYY.DDD
```

Figure 65: Define an IAM Alternate Index Panel

After filling in the necessary information with the desired values, press enter. IAM will return to the IAM Primary Option Menu, and indicate the status of the define request in the upper right corner. To display the file characteristics, make sure that the desired dataset name is filled in, then press enter.

46.02 CONTINUED . . .

IAM AIX CHARACTERISTICS PANEL

To display the alternate index file characteristics start at the IAM Primary Options Panel. Leave the OPTION field blank, and fill in the alternate index dataset name field, then press enter. Shown below is the IAM Alternate Index File characteristics panel for the alternate index that was just defined.

```
---------------- IAM Alternate Index File Characteristics -----------------
 COMMAND ===>
 Dataset Name: RAM2.MYIAM.AIX

 Definition Information            Allocation Information
    Record Length:        73          Volume:              JUNK01
    Record Format:  VARIABLE          Device Type:           3380
    Key Length:            4          Tracks in use:            0
    Key Offset:            5          Block Size:            4096
    Key Offset(BASE):     16          Blocking Factor:       4096
    Unique Keys:          NO          Alloc Type:       CYLINDERS
    Share Option:          2          Primary Alloc:            2
    Release:             YES          Secondary Alloc:          1
    Storage Class:
    Data Class:
    Mgmt Class:

 Statistics                        Extended Area Information
    Creation:       2002.218          Overflow Records:         0
    Expiration:     0000.000          Blocks Allocated:         0
    Last Reference: 2002.218          Blocks Used:              0
    Records:               0          Overflow Blocks:          0
    Deletes:               0          PE Blocks:                0
    Inserts:               0          Blocks Available:         0
    Updates:               0          Variable Overflow:      YES
    Minimum Buffers:
    Maximum Buffers:

 Associations
    CLUSTER-RAM2.MYIAM.CLUSTER
        AIX-RAM2.MYIAM.AIX
```

Figure 66: Example of IAMISPF File Characteristics display for an Enhanced Format File

BUILDING THE ALTERNATE INDEX

So far, we have defined and loaded the base IAM cluster, and defined the alternate index. Our next step is to build the alternate index. From the IAM Primary Options Panel, type a **U** for the option to bring up the IAM utilities panel, and press enter. When the IAM Utility Program Menu is displayed, enter an **I** for the option to specify the BLDINDEX function, as shown below:

```
-------------------- IAM UTILITY PROGRAM SELECTION MENU --------------------
 OPTION ===> i

    S  Specify utility print dataset allocation parameters

    I  BLDINDEX - Build Alternate Index
    R  IAMRECVR - IAM file diagnostic and recovery utility
    V  IAMSTATS - IAM VSAM Interface (VIF) module information
    X  IAMXMONI - IAM Execution Monitor/ISPF for CICS Regions
    Z  IAMZAPOP - IAM options table utility
```

Figure 67: Selecting BLDINDEX on the IAM Utility Program Selection Menu

46.02 CONTINUED . . .

BUILD INDEX PANEL On the IAM BLDINDEX panel, enter the names of the base cluster and the alternate index to be built. When ready, press enter to start the BLDINDEX function.

```
----------------------------- BLDINDEX - Datasets -----------------------------
COMMAND ===>

   Base Cluster ===> myiam.cluster

Alternate Index ===> myiam.aix

Specify BASE and AIX dataset names and hit ENTER
```

Figure 68: Example of completed BLDINDEX Panel

After pressing enter, the IAM ISPF panels will invoke the IDCAMS BLDINDEX function to build the alternate index. When completed, IAM ISPF will display the IDCAMS output, an example of which is shown below.

```
BROWSE    SYS02321.T145744.RA000.RAM2.R0100508                        Line 00000000 Col 001 120
Command ===>                                                                    Scroll ===> CSR
*****************************************Top of Data*********************************************
IDCAMS  SYSTEM SERVICES                         TIME: 14:57:45        11/17/02     PAGE      1
BLDINDEX  INFILE(DISKIN) OUTFILE(DISKOUT)
IDC0652I RAM2.MYIAM.AIX SUCCESSFULLY BUILT
IDC0001I FUNCTION COMPLETED, HIGHEST CONDITION CODE WAS 0
IDC0002I IDCAMS PROCESSING COMPLETE. MAXIMUM CONDITION CODE WAS 0
************************************ Bottom of Data *********************************************
```

Figure 69: Example of Output from BLDINDEX Function

DEFINE AN IAM PATH The next step is to define an IAM PATH. Here again, you start out with the IAM Primary Options Panel. The procedure is as follows:

- Type an **I** in the Option field, to define an IAM dataset.
- Type in the PATH name in the dataset name field.
- Type a **P** in dataset type for a Path.
- Press Enter.

DEFINE AN IAM PATH PANEL The IAM Define Path panel will be displayed. On this panel you need to specify the name of the path entry dataset, which will typically be the name of an alternate index. Then indicate whether or not any IAM is to automatically update the upgradeable alternate index datasets when this path is opened for update, by specifying YES or NO to the UPDATE field. Then press enter. An example of a completed Define an IAM Path panel is shown below.

```
----------------------------- DEFINE AN IAM PATH -----------------------------
COMMAND ===>

 Path Name ===> RAM2.MYIAM.PATH
Path Entry ===> ram2.myiam.aix

    Update ===> YES  Yes = Open all UPGRADE sets when PATH opens
                      No = Open only BASE CLUSTER when PATH opens
```

Figure 70: Example of the Define an IAM Path Panel

46.02 CONTINUED . . .

IAM PATH CHARACTERISTICS PANEL

When the path definition is completed, IAM ISPF will redisplay the IAM Primary Options panel, with a message in the upper right hand corner indicating the status of the define request. To display the path characteristics, just hit enter again.

```
------------------------ IAM Path Characteristics ------------------------
COMMAND ===>
Dataset Name: RAM2.MYIAM.PATH

Definition Information            Allocation Information
   Base Record Length:   128         Volume:              JUNK01
   AIX Key Offset:        16         Device Type:           3380
   AIX Key Length          4         Tracks in use:            1
   Update:                NO         Blocking Factor:          4
                                     Alloc Type:         RECORDS

Statistics
   Creation:       2002.218
   Expiration:     0000.000
   Last Reference: 2002.218

Associations
   CLUSTER-RAM2.MYIAM.CLUSTER
       AIX-RAM2.MYIAM.AIX
      PATH-RAM2.MYIAM.PATH
```

Figure 71: IAM Path Characteristics Panel

DEFINE A VSAM CLUSTER

The IAM ISPF panels also offer the capability to define VSAM datasets, in a manner similar to the IAM file definition. Starting from the IAM Primary Options Panel, do the following:

- Type a **V** for VSAM Define in the OPTION field.
- Enter the dataset name for the VSAM cluster in the dataset name field.
- Indicate type of dataset, such as **C** for cluster.
- Optionally, type in the name of an IAM or VSAM dataset with similar attributes in the Model field.
- Press the enter key.

Shown below is a completed IAM Primary Options Panel for a VSAM dataset definition.

```
   ------------------------- IAM PRIMARY OPTION MENU -------------------------
OPTION ===> v

   I      - Allocate (DEFINE) a new IAM Dataset             Ver 8.0/01P
   V      - Allocate (DEFINE) a new VSAM Cluster
   D      - Delete a Dataset, Cluster, Path, or Alternate Index
   C      - Copy an IAM Dataset, VSAM KSDS, or VSAM ESDS
   M      - Move an IAM Dataset, VSAM KSDS, or VSAM ESDS
   R      - Rename a Dataset, Cluster, Path, or Alternate Index
   U      - Invoke an IAM Utility program
   blank  - Display IAM Dataset or VSAM Cluster Information


Enter dataset name (required for all options except U)
 Dataset Name       ===> myvsam.cluster
 Dataset Type       ===> C  C=Cluster X=AIX P=Path (options I, V Only)

    Enter Model or New dsname (optional for options I and V, required for option R)
 Model|Newname      ===>


Delete Confirmation ===> YES  Yes|No
```

Figure 72: IAM Primary Options Panel set for VSAM Define

46.02 CONTINUED . . .

VSAM DEFINE PANEL

After pressing enter, the following VSAM Define panel is displayed. This panel can be used to define a VSAM ESDS, KSDS, LDS, or RRDS. To define a multi-volume cluster, specify YES in the Multi-volume Allocation field. An additional panel will be displayed where the additional volumes can be entered. IAM files can also be defined using this panel by setting the OWNER parameter to $IAM, however the IAM overrides are not available on this panel, as they are on the IAM Define Panel. Fill in the necessary information for the data component of the VSAM cluster, then press enter. An example of a completed panel is displayed below.

```
---------------------------- DEFINE A VSAM CLUSTER -----------------------------
 COMMAND ===>

 Cluster Name :  RAM2.MYVSAM.CLUSTER
 Data Name ===> 'RAM2.MYVSAM.CLUSTER.DATA'

 ALLOCATION                                 Multi-Volume Allocation ===> NO
  Volume              ===> junk01
  SMS Storage Class   ===>                  MISC. ATTRIBUTES
  SMS Data Class      ===>                   Owner              ===>
  SMS Mgmt Class      ===>                   X Region Share     ===> 2        1-4
  Cyls|Recs|Trks      ===> cyls              X System Share     ===> 3        3-4
  Primary Space       ===> 2                 Bufferspace        ===>
  Secondary Space     ===> 1                 Reuse              ===> no       Yes|No
 ATTRIBUTES                                  Speed              ===> yes      Yes|No
  ESDS|KSDS|RRDS|LDS  ===> ksds              Spanned            ===>          Yes|No
  Max Recordsize      ===> 128               Erase              ===>          Yes|No
  Avg Recordsize      ===> 100
  Key Length          ===> 4
  Key Offset          ===> 8
  CI Size             ===> 4096             RETENTION
  CI Freespace %      ===> 10                DAYS               ===>          0-9999
  CA Freespace %      ===> 5                 EXPIRATION DATE    ===>          YYYY.DDD
```

Figure 73: Sample IAM Panel to Define a VSAM Cluster

INDEX ALLOCATION FOR A KSDS

If you are defining a VSAM KSDS, then the following panel will be. This panel can be used to provide index specific parameters. Complete the necessary information on this panel, then press enter. A sample of a completed panel is shown below.

```
----------------------- INDEX ALLOCATION FOR A VSAM KSDS -----------------------
 COMMAND ===>

 Cluster Name   :  RAM2.MYVSAM.CLUSTER
 Data Name      :  RAM2.MYVSAM.CLUSTER.DATA
 Index Name ===>  'RAM2.MYVSAM.CLUSTER.INDEX'

 ALLOCATION
  Volume(s)           ===> junk01   ______   ______   ______   ______
  Cyls|Recs|Trks      ===> trks
  Primary Space       ===> 2
  Secondary Space     ===> 1

 ATTRIBUTES
  CI Size             ===> 512
  Imbed               ===> no        Yes|No
  Replicate           ===> no        Yes|No
```

Figure 74: Example of VSAM Index Component Define

After pressing enter, IAM will define the dataset, then return to the IAM Primary Options Panel. The status of the define will appear in the upper right hand corner of the panel.

46.02 CONTINUED . . .

COPY FROM IAM TO VSAM

Next, the VSAM cluster can be loaded by copying from a dataset, such as an IAM file, into the VSAM dataset. By starting on the IAM Primary Options Panel, do the following:

- Type **C** in the Option field for Copy function.
- Enter the name of the source dataset in the dataset name field.
- Press the enter key.

Shown below is an example of the completed Primary Options Panel for a copy.

```
-------------------------- IAM PRIMARY OPTION MENU -------- DATASET ALLOCATED
 OPTION ===> c

    I      - Allocate (DEFINE) a new IAM Dataset                    Ver DLEV-8.0
    V      - Allocate (DEFINE) a new VSAM Cluster
    D      - Delete a Dataset, Cluster, Path, or Alternate Index
    C      - Copy an IAM Dataset, VSAM KSDS, or VSAM ESDS
    M      - Move an IAM Dataset, VSAM KSDS, or VSAM ESDS
    R      - Rename a Dataset, Cluster, Path, or Alternate Index
    U      - Invoke an IAM Utility program
    blank - Display IAM Dataset or VSAM Cluster Information


 Enter dataset name (required for all options except U)
  Dataset Name       ===> myiam.cluster
  Dataset Type       ===> C  C=Cluster X=AIX P=Path (options I, V Only)

 Enter Model or New dsname (optional for options I and V, required for option R)
  Model|Newname      ===>


 Delete Confirmation ===> YES  Yes|No
```

Figure 75: Example of Primary Options set for COPY

After pressing enter, IAM will display the panel for the target dataset. Here we enter the name of the VSAM dataset in the Target dataset name field, then press enter to perform the copy.

```
--------------- COPY AN IAM DATASET, VSAM ESDS, OR VSAM KSDS ----------------
 COMMAND ===>

 Source IAM Dataset Name:   RAM2.MYIAM.CLUSTER


 Enter target dataset or cluster name
   Target Dataset Name ===> myvsam.cluster
   Extended Copy       ===> NO    (Yes|No    - To define copy selection criteria)


 Define Target Dataset ===> NO    (Yes|No    - If target does not exist)
 Target Dataset Type   ===>       (IAM|VSAM - If target is to be defined)
```

Figure 76: Example of Copy Target Specification Panel

46.02 CONTINUED . . .

VSAM DATA COMPONENT INFORMATION

The IAM ISPF panels also have a VSAM data component information panel that is shown below. This is a condensed display of the most relevant fields from the IDCAMS LISTCAT command. For multi-volume data components, an additional panel will be displayed which contains all the volumes the data component has been allocated on. To display information on a VSAM cluster, start at the IAM Primary Options panel. Leave the Option field blank, and fill in the DataSet Name field with the name of the cluster you would like to retrieve information about. Then, press enter. The following is an example of the resulting display.

```
----------------------- VSAM DATA COMPONENT INFORMATION -----------------------
 COMMAND ===>

 Cluster Name: RAM2.MYVSAM.CLUSTER
 Data Name:    RAM2.MYVSAM.CLUSTER.DATA

 ATTRIBUTES                ALLOCATION                  STATISTICS
  Dataset Type:      KSDS   Volume:          JUNK01     Records:          10000
  CI Size:           4096   Device Type:       3380     Deletes:              0
  CI's per CA:        150   Alloc Type:   CYLINDERS     Inserts:              0
  Avg Recordsize:     100   Primary Alloc:        2     Updates:              0
  Max Recordsize:     128   Secondary Alloc:      1     Retrievals:           0
  Key Length:           4   Tracks Alloc:        45     Excps:               30
  Key Offset:           8   Tracks Used:         45     CI Splits:            0
  Cross Region Shropt:  2   Extents:              2     CA Splits:            0
  Cross System Shropt:  3   Storage Class:
  Bufferspace:       9728   Data Class:                FREE SPACE
  Reuse:               NO   Mgmt Class:                 CI Freespace %:      10
  Speed:              YES                               CA Freespace %:       5
  Spanned:             NO  CLUSTER HISTORY              Bytes:                0
  Erase:               NO   Owner:                      Tracks:               0
                            Creation:       2002.322
                            Expiration:     0000.000
```

Figure 77: Example of VSAM Data Component Information Panel

VSAM INDEX COMPONENT INFORMATION

For VSAM KSDS files there is an additional VSAM index component information panel. This panel will be displayed by hitting enter from the data component information panel, and will only be displayed for VSAM KSDS's. This is condensed display of the most relevant fields from the IDCAMS LISTCAT command. For multi-volume index components, an additional panel will be displayed which contains all the volumes the index component has been allocated on.

```
----------------------- VSAM INDEX COMPONENT INFORMATION ----------------------
 COMMAND ===>

 Cluster Name: RAM2.MYVSAM.CLUSTER
 Index Name:   RAM2.MYVSAM.CLUSTER.INDEX

 ATTRIBUTES                ALLOCATION                  STATISTICS
  CI Size:           1536   Volume:          JUNK01     Records:              4
  CI's per CA:         23   Device Type:       3380     Deletes:              0
  Max Recordsize:    1529   Alloc Type:      TRACKS     Inserts:              0
  Key Length:           4   Primary Alloc:        2     Updates:              0
  Key Offset:           8   Secondary Alloc:      1     Retrievals:           0
  Imbed:               NO   Tracks Alloc:         2     Excps:               14
  Replicate:           NO   Tracks Used:          1     CI Splits:            0
                            Seq-Set Tracks:       0     CA Splits:            0
 CLUSTER HISTORY            Total Used:           1     Entries/Section:     12
  Owner:                    Extents:              1     Index Levels:         2
  Creation:      2002.322   Storage Class:              Seq-Set RBA:          0
  Expiration:    0000.000   Data Class:                 Hi-Level RBA:      3072
                   Mgmt Class:
```

Figure 78: Example of VSAM Index Component Information Panel

46.02 CONTINUED . . .

IAM ISPF RENAME FUNCTION

The IAM ISPF panels also have a RENAME function, that will rename IAM or VSAM datasets. Starting at the IAM Primary Options Panel, do the following:

- Type an **R** in the Option field.
- Enter the name of the dataset to be renamed in the dataset name field.
- Enter the new name of the dataset in the NEWNAME field.
- Press enter.

Shown below is a completed Primary Options panel to rename an IAM dataset.

```
-------------------------- IAM PRIMARY OPTION MENU ---------- COPY SUCCESSFUL
 OPTION ===> r

    I      - Allocate (DEFINE) a new IAM Dataset                    Ver 8.0/01P
    V      - Allocate (DEFINE) a new VSAM Cluster
    D      - Delete a Dataset, Cluster, Path, or Alternate Index
    C      - Copy an IAM Dataset, VSAM KSDS, or VSAM ESDS
    M      - Move an IAM Dataset, VSAM KSDS, or VSAM ESDS
    R      - Rename a Dataset, Cluster, Path, or Alternate Index
    U      - Invoke an IAM Utility program
    blank  - Display IAM Dataset or VSAM Cluster Information


 Enter dataset name (required for all options except U)
  Dataset Name        ===> testiam.cluster
  Dataset Type        ===> C  C=Cluster X=AIX P=Path (options I, V Only)

 Enter Model or New dsname (optional for options I and V, required for option R)
  Model|Newname       ===> oldiam.cluster


 Delete Confirmation ===> YES  Yes|No
```

Figure 79: Completed Primary Options Panel for Rename

After pressing enter, for most dataset types the IAM Primary Option panel is redisplayed, with the message DATASET RENAMED in the upper right hand corner. However, if the dataset is an IAM dataset that is part of an alternate index association, then the following panel will appear to cause IAM to automatically recatalog the dataset to update all of the related associations. Make sure that the RECATALOG field is set to YES, then press enter.

```
----------------------------- RECATALOG Components ----------- DATASET RENAMED
 COMMAND ===>

    New Dataset Name... RAM2.NEWIAM.CLUSTER

    The above dataset that has been renamed is associated with an IAM
    Alternate Index or Path and needs to be RECATALOGED so IAM can
    maintain the correct relationships with associated datasets.

    Perform RECATALOG ===> YES  (Yes/No)
```

Figure 80: Sample RECATALOG Confirmation Panel

46.02 CONTINUED . . .

DELETING DATASETS The IAM ISPF panels also offer a delete function for IAM and VSAM datasets. For datasets with alternate index and path associations, the delete may cause additional datasets to be deleted. For example, the delete of the base cluster will cause all associated alternate index datasets and paths to be deleted. The deletion of an alternate index dataset will cause the deletion of any associated paths. From the IAM Primary Options Panel, enter the following:

- Type a **D** in the option field for delete.
- Enter the dataset name in the dataset name field.
- Set Delete Confirmation to YES or NO. It is recommended that YES be used, which will cause a delete confirmation panel to be displayed.
- Press enter.

Shown below is an example of the completed primary options panel for a delete.

```
--------------------------- IAM PRIMARY OPTION MENU ---------------------------
OPTION ===> d

   I     - Allocate (DEFINE) a new IAM Dataset                   Ver 8.0/01P
   V     - Allocate (DEFINE) a new VSAM Cluster
   D     - Delete a Dataset, Cluster, Path, or Alternate Index
   C     - Copy an IAM Dataset, VSAM KSDS, or VSAM ESDS
   M     - Move an IAM Dataset, VSAM KSDS, or VSAM ESDS
   R     - Rename a Dataset, Cluster, Path, or Alternate Index
   U     - Invoke an IAM Utility program
   blank - Display IAM Dataset or VSAM Cluster Information


Enter dataset name (required for all options except U)
 Dataset Name      ===> myiam.cluster
 Dataset Type      ===> C  C=Cluster X=AIX P=Path (options I, V Only)

Enter Model or New dsname (optional for options I and V, required for option R)
 Model|Newname     ===>


Delete Confirmation ===> YES  Yes|No
```

Figure 81: Completed Primary Options Panel for Delete

46.02 CONTINUED . . .

DELETE CONFIRMATION

IAM ISPF will then display the following Delete Confirmation panel. This will include a list of associated datasets from an alternate index sphere that will also be deleted. However, not all of the possible associations will be shown, for example paths to the alternate indexes which will be deleted will not be displayed when deleting a base cluster. To cancel the delete request enter END (or generally press PF3). To allow the dataset delete to be performed, press enter.

```
--------------------------- CONFIRM DELETE REQUEST ---------- Row 1 to 1 of 1
 COMMAND ===>                                                  SCROLL===> PAGE

 DATASET NAME:     RAM2.MYIAM.CLUSTER
 DATASET TYPE:     IAM KSDS
 VOLUME:           JUNK01
 CREATION DATE:    2000.328
 EXPIRATION DATE:  0000.000

    Press ENTER to confirm the delete request.

    Enter END command to cancel the delete request.


 The following related objects will also be deleted

 TYPE NAME
 ---- --------------------------------------------
 (G)  RAM2.MYIAM.AIX
******************************* Bottom of data ********************************
```

Figure 82: DataSet Delete Confirmation Panel

Upon completion of the delete process, the Delete Request Results panel will be displayed, which will include a list of all of the dataset and related components that were deleted.

```
---------------------------  DELETE REQUEST RESULTS---------- Row 1 to 3 of 3
 COMMAND ===>                                                  SCROLL===> PAGE

 DATASET NAME:     RAM2.MYIAM.CLUSTER
 DATASET TYPE:     IAM KSDS
 VOLUME:           JUNK01
 CREATION DATE:    2000.328
 EXPIRATION DATE:  0000.000

    Press ENTER to continue

 The following objects have been DELETED

 TYPE NAME
 ---- --------------------------------------------
 (A)  RAM2.MYIAM.PATH
 (A)  RAM2.MYIAM.AIX
 (A)  RAM2.MYIAM.CLUSTER
******************************* Bottom of data ********************************
```

Figure 83: Example of the Delete Request Results Panel

After pressing enter again, the IAM Primary Options panel will be displayed, with a message indicating the status of the delete request.

46.02 CONTINUED . . .

IAM UTILITY PROGRAM SELECTION MENU

The IAM ISPF panels support some limited access to some of the IAM utilities through the IAM utility selection panel. Since all the IAM utilities are really batch utilities, the additional panels simply build and pass an in storage command stream to the requested utility. The utility output can be directed to either a temporary print dataset or a new print dataset. ISPF BROWSE is called to view the utility print output after the utility executes.

```
--------------------- IAM UTILITY PROGRAM SELECTION MENU --------------------
OPTION ===>

   S  Specify utility print dataset allocation parameters

   I  BLDINDEX - Build Alternate Index
   R  IAMRECVR - IAM file diagnostic and recovery utility
   V  IAMSTATS - IAM VSAM Interface (VIF) module information
   X  IAMXMONI - IAM Execution Monitor/ISPF for CICS Regions
   Z  IAMZAPOP - IAM options table utility
```

Figure 84: IAM Utility Selection Menu

IAM UTILITY PRINT DATASET ALLOCATION PANEL

The utility print dataset allocation parms panel specifies the allocation for the output dataset from the utilities. The utility print datasets will be allocated using the values specified in this panel. For most utilities, the default parameters should be sufficient. However, if the IAMRECVR PRINT function is used, you may wish to use this panel to provide additional space since this function can generate a significant amount of output depending on the options selected.

```
----------------- IAM UTILITY PRINT DATASET ALLOCATION PANEL ----------------
COMMAND ===> _

Blks|Cyls|Trks    ===> CYLINDERS
Primary Space     ===> 1
Secondary Space   ===> 1

Note - If blocks is specified, an average blksize of 6171 will be used
```

Figure 85: Example of Utility Print DataSet Allocation

IAMRECVR FUNCTION SELECTION MENU

The IAM IAMRECVR function selection panel is shown below. For execution under ISPF, the DIAGNOSE, LIST, and PRINT functions are supported. The other IAMRECVR functions are only supported in batch.

```
---------------------- IAMRECVR FUNCTION SELECTION MENU ---------------------
OPTION ===> _

   D  DIAGNOSE - Scan an IAM file for errors
   L  LIST     - List file characteristics
   P  PRINT    - Dump selected areas or blocks from an IAM file
```

Figure 86: IAMRECVR Function Selection Menu

IAMRECVR DIAGNOSE OPTIONS MENU

Shown below is the IAMRECVR DIAGNOSE options menu. DIAGNOSE can be used to validate the structural integrity of an IAM file. As with all of the utility option panels, if you leave the print DSN field blank, the print output will be directed to a temporary print dataset. If you provide a print dataset name, the name must be new name. The print dataset will be allocated for you.

```
---------------------- IAMRECVR - DIAGNOSE OPTIONS MENU ---------------------
COMMAND ===> _

Dataset Name ===> 'IAMV.MARK$IAM.IAMFILE1'
Print DSN    ===>             (optional)
BSAM         ===> NO          (Yes|No - Use BSAM instead of EXCP)
```

Figure 87: Example of IAMRECVR DIAGNOSE Options Menu

46.02 CONTINUED . . .

IAMRECVR LIST OPTIONS MENU

Shown below is the IAMRECVR LIST OPTIONS menu. The LIST function produces output similar to the IAMPRINT report provided by the IDCAMS LISTCAT command.

```
------------------------ IAMRECVR - LIST OPTIONS MENU ------------------------
COMMAND ===> _

Dataset Name ===> 'IAMV.MARK$IAM.IAMFILE1'
Print DSN    ===>                           (optional)
```

Figure 88: Example of IAMRECVR List Options Menu

IAMRECVR PRINT OPTIONS MENU

Shown below is the IAMRECVR PRINT OPTIONS menu. The PRINT function produces a formatted dump of areas you specify within an IAM file.

```
------------------------ IAMRECVR - PRINT OPTIONS MENU ------------------------
COMMAND ===> _

Dataset Name ===> 'IAMV.MARK$IAM.IAMFILE1'
Print DSN    ===>                                 (optional)
BSAM         ===> NO        (YesINo - Use BSAM instead of EXCP)
IDPINQ       ===> NO        (YesINo - Print IAM control block)
PRINT LENGTH ===>           (32-32768 - Maximum data to print per block)
MAX BLOCKS   ===>           (Maximum blocks to print from each area)
ALL BLOCKS   ===>           (YesINo - Print blocks from all areas)
              OR
DATA         ===>           (YesINo - Print data blocks)
KEYS         ===>           (YesINo - Print key blocks)
OVERFLOW     ===>           (YesINo - Print overflow blocks)
PE           ===>           (YesINo - Print prime extension blocks)
              OR
FROM BLOCK   ===>           (First block number to print)
TO BLOCK     ===>           (Last  block number to print)
```

Figure 89: Example of IAMRECVR Print Options Menu

IAMZAPOP FUNCTION SELECTION MENU

Shown below is the IAMZAPOP function selection menu. The ZAP function is only supported in batch and not under ISPF.

The LIST function produces a report of the options from the IAM Global Options Table in the IAM library specified.

The RESET function is used to reset the IAM Global Options Table in the library specified back to its distributed default values.

The COPY function will copy the IAM Global Options Table settings from the IAM Library Name specified to the IAM Global Options Table in the COPY to DSNAME specified. Only the changed options will be copied to the new table.

The AUDIT function will list all of the options in the IAM Global Options Table that have been changed from their distributed default values.

You may provide the name of an optional new print dataset. If this field is left blank, a temporary dataset will be used.

```
----------------------- IAMZAPOP FUNCTION SELECTION MENU ----------------------
OPTION ===> _

   L  LIST     - List IAM options
   R  RESET    - Reset IAM options
   C  COPY     - COPY IAM options
   A  AUDIT    - AUDIT IAM options

 IAM Library Name ===>
Print DSNAME     ===>                        (optional)
COPY to DSNAME   ===>                        (for copy)
```

Figure 90: Example of IAMZAPOP Function Selection Menu

46.02 CONTINUED . . .

IAMXMONI CICS FILE STATISTICS

The IAM ISPF panels also offer the capability to retrieve file activity information for IAM files that are being used under CICS under TSO. This is the same information that is available under CICS by using the IAMXMON transaction. To use this capability, you must be logged on to the same CPU/ LPAR that CICS is running under. Type **X** in the option field for the IAM Utility panel, and press enter. IAM ISPF will display the following screen:

```
Innovation Access Method (IAM) Execution Monitor 8.0/01P
 Command ===>

         MVS JOBNAME of CICS Region to Monitor ===> GFMTS120


   Display DDNAMEs that begin with ===>
   IAM File types to display       ===> ALL      (ALL, KSDS, ESDS, PATH)
   DSN list initial sort field     ===> DDNAME   (DDNAME, DSN, STG)
```

Figure 91: Example of IAM ISPF IAMXMON Panel

On the above panel, you must fill in the JOB NAME or started task name of the CICS region from which you want statistical information. There are also a few options that can be specified to customize the output panel. Once the panel is completed as desired, press enter. IAM will then display a summary of the IAM Enhanced format files that are open under CICS, as seen below.

IAMXMON DATASET SUMMARY PANEL

```
Innovation Access Method (IAM) Execution Monitor 8.0/01P       Row 1 to 3 of 3
 Command ===>

   Region: GFMTS120
   Storage used for displayed IAM files:      1 MB     Dataspace Size:      0 MB
   Above 16MB Storage Limit for Region:      32 MB     Dataspace Used:      0 MB

   Status Indicators:  K KSDS File     U Open for Update     R REORG Suggested
                       E ESDS File     I Open for Input      S DSN Shared

   DDNAME    DSN                                         Storage(K)  Status
   --------  ------------------------------------------  ----------  --------
 . AIXPATH   GFM.GAMA.KSDS2.PATH                                 12  P U
 . SYS00002  GFM.GAMA.KSDS2.BASE                                 76  KBU
 . SYS00003  GFM.GAMA.KSDS2.AIX                                  92  KXU
 ******************************* Bottom of data ********************************
```

Figure 92: Example of IAMXMON DataSet Summary Panel

For IAM datasets that are opened under multiple DD names and that are sharing the same IAM control block structure will appear only under one of the DD names, which will be the one IAM is actively using at that point in time. IAM datasets that are opened as a result of an alternate index association will have DD names that begin with SYS followed by five numeric digits. To obtain more data about any particular dataset, move the cursor to the dataset that you want information on, and press enter. IAM will then display detailed statistical information, as shown in the following example.

46.02 CONTINUED . . .

IAMXMON DATASET DETAIL PANEL

```
Innovation Access Method (IAM) Execution Monitor 8.0/00T
 COMMAND ===>
 DDNAME: SYS00002   DSN: GFM.GAMA.KSDS2.BASE
 Region: GFMTS120

 IAM Command Execution Summary
  GET Random...... 00000000   PUT Update....... 00000000   ENDREQ...... 00000000
  GET Sequential.. 00000000   PUT Add.......... 00000000   WRTBFR...... 00000000
  GET Previous.... 00000000   Point............ 00000000   OPEN........ 00000001
  GET KGE......... 00000000   Point KGE........ 00000000   CLOSE....... 00000000
  GET (Skip)...... 00000000   ERASE............ 00000000   CLOSE T..... 00000000
  Verify.......... 00000000   Invalid Req...... 00000000   Len Change.. 00000000
  Seq Chain Read.. 00000000   Seq Chain Write.. 00000000   Flush Bfr... 00000000

 IAM Execution Statistics                    Total Records............ 0000000012
   Requests Processed......... 000000000   Requests Failed........... 000000000
   Blocks Read................ 000000002   Blocks Written............ 000000000
   Dynamic Table Retrievals... 000000000   Dynamic Table Records..... 000000000
   Dynamic Buffer Retrievals.. 000000000   Minimum Buffers................ 0001
   Storage Above 16M.......... 000073728   Maximum Buffers................ 0078
   Storage Below 16M.......... 000004096   Current Buffers................ 0004

 Extended Overflow Information
   Available Overflow Blocks.. 000000000  Extended PE Blocks Used.... 000000000
   Extended Data Blocks....... 000000000  Extended Data Records...... 000000000

 Controlling Program... DFHKETCB

 Processing Options                  Dataset Attributes
   OPENed for UPDATE.. YES             Dataset Type.. KSDS  Record Size... 84
   Immediate WRITE.... NO              Shareoptions.. 2     Block Size.... 11476
   OPTCD=ASY.......... NO              Extents....... 1     Recs/Block.... 136
   OPTCD=WAITX........ NO              Blks/Trk...... 4     Record Fmt.... V
   OPTCD=WRITE........ NO              Data Comp..... NO    Index Comp.... NO
   CI Processing...... NO              Key Length.... 8     RKP........... 4
   JRNAD Loaded....... NO              Integrated Pct 0
   JRNAD Active....... NO
   UPAD Loaded........ NO
   UPAD Active........ NO
```

Figure 93: Example of IAMXMON DataSet Detail Panel

IAMXMON PATH PANEL The detail panel for an IAM path is similar to the summary panel. The Path Detail Panel will display the datasets that are the components of the path, from which individual datasets can be selected similar to the dataset summary panel. An example of a path panel is shown below.

```
 Innovation Access Method (IAM) Execution Monitor 8.0/00T       Row 1 to 2 of 2
  Command ===>

    Path DD: AIXPATH    Path Name: GFM.GAMA.KSDS2.PATH
     Region: GFMTS120

    TYPE  DDNAME    DSN   Storage                                   Storage(K)
    ----  --------  ------------------------------------------- ----------
  . BASE  SYS00002  GFM.GAMA.KSDS2.BASE                                     76
  . AIX   SYS00003  GFM.GAMA.KSDS2.AIX                                      92
  ****************************** Bottom of data *******************************
```

Figure 94: Example of an IAMXMON Path Panel

46.02 CONTINUED . . .

RESULTS OF IAMSTATS REQUEST

Shown below is an example of the results of the IAMSTATS request. The IAM version shows the version of the IAM options table that was found in core. The VIF key is an internal indicator used to allow multiple versions of IAM to run concurrently and not interfere with each other. There is a different VIF key for each IAM version, and sometimes a different key for different releases within the same version.

The IAM Vector Table status can be INSTALLED or NOT INSTALLED. If the status is NOT INSTALLED, the job used at your site to start IAM has NOT been run. The status of IAM itself can be ACTIVE or NOT ACTIVE. If the vector table is NOT INSTALLED, the status will always be NOT ACTIVE. The status may also be NOT ACTIVE if IAM has been started, then stopped. Stopping IAM does not remove the IAM vector table from memory.

The rest of the display lists all the IAM modules that are resident, where they are currently loaded, their length and their version/level information if available. If there are multiple versions and/or levels of IAM active, hitting the ENTER key will display information about each IAM vector table found.

```
-------------------- IAM VSAM Interface (VIF) Statistics --------------------
OPTION ===>
                                                 Module    Address    Len     Version
 IAM Version: 8.0                                IAMVECTB  00C938B0  000750   8.0/01P
      VIFKEY: FFFE0050                           IAM0192A  00C88470  001B90   8.0/01P
                                                 IAM0200T  00C90128  000ED8   8.0/01P
                                                 IAM0231T  00C86AA8  001558   8.0/01P
 IAM Vector Table is INSTALLED                   IGG0CLA0  00C933A8  000508
 and IAM is ACTIVE                               IAM0CLA0  80C8D388  000C78   8.0/01P
                                                 IAM026DU  85B156A0  004960   8.0/01P
                                                 IAMCSMF   00C91988  000678   8.0/01P
 Test Version of IAM: NO                         IAM0009I  86379730  0004A0   8.0/01P
           Jobnames:                             IAM0001I  85B13B60  0014A0   8.0/01P
                                                 IAM00020  85BEC3F0  000C10   8.0/01P
 IAM AIX Support Active: YES                     IGWABWO   863CF648  000220
                                                 IAMABWO   86379240  0004F0   8.0/01P
                                                 IGWARLS   863CFB10  000258
                                                 IAMARLS   85BE9AB0  000550   8.0/01P
```

Figure 95: Example of IAM Stats Output

THIS PAGE WAS INTENTIONALLY LEFT BLANK

47.01 IAMJREST - IAM JOURNAL RESTORE PROGRAM OVERVIEW

OVERVIEW IAMJREST is used to restore an IAM dataset to its desired state using the journal records collected by the optional IAM journal exit. IAMJREST can use the IAM log datasets produced by non-RLS file updates or the IAM RLS log datasets. The IAM journal exit is activated by use of the IAM Override facility, by specifying the JRNAD keyword. IAM journaling is described in Section 10.88 IAM Journal and Recovery for non-RLS processing, and in Section 20.30 IAM RLS Journaling.

In the IAM RLS environment, IAMJREST, along with the locking services of IAM RLS, provides a recovery mechanism that can be used to backout updates made by an abending job step, while retaining the updates to other records made by concurrent jobs. IAMJREST can also be used as part of a forward recovery process, when an IAM dataset had to be restored. In this case, IAMJREST can reapply all the updates to the restored file that occurred since it was backed up.

The use of these capabilities in a non-RLS environment can be helpful to improve data availability by reducing the frequency of backing up the complete dataset, and enhancing recovery after a batch job abends. For example, rather than backing up the complete dataset daily, the complete dataset can be backed up less frequently, while backing up the IAM journal dataset on a daily basis. This provides the same level of recovery capability, while reducing the backup time presuming that the amount of update activity is low to moderate. Recovery from batch job abends can also be improved by using the ability to backout the file updates that were done by the job step(s), as opposed to restoring the dataset and rerunning file updates to get back to where the failure occurred.

The capabilities of IAMJREST include:

- Perform a forward recovery. After an IAM dataset has been restored from a backup copy, IAMJREST can be used to apply all of the updates to the dataset from the journal file.
- Perform a backout (or backwards) recovery. This type of recovery will back out changes from the specified job step(s), to restore a dataset to an image prior to the start of the desired job or job step.
- Perform a backout recovery for multiple datasets that were being accessed under IAM RLS for failed batch jobs. If the failed batch job has been performing IAM batch syncpoints, then the backout can optionally be performed to the most recent syncpoint prior to the abend.

47.01 CONTINUED . . .

FORWARD RECOVERY PROCESS

A forward recovery is a process that will update a file with the updated records that are selected from the journal. The forward recovery process begins by independently restoring the file's contents from a backup copy. After a successful restore, the file is updated using the IAMJREST RESTORE FORWARD command with all of the updated records that are on the journal, up to the specified point in time. Actually, IAMJREST will sort the selected journal records by key (or RBA for ESDS files) and the time stamp. Then IAMJREST will only apply to the IAM file the most recent update for each record.

To be able to perform a forward recovery, the journal file must contain after images. This is accomplished by specifying either JRNAD=AFTER or JRNAD=BOTH on the IAM CREATE override. To be able to perform a forward recovery encompassing all of the file updates, then the IAM journalling must be specified when the file is defined or loaded using the IAM CREATE override. This way there is not a concern about having missed specifying journalling on any of the update jobs.

For a typical journal set up to perform forward recoveries, the journal dataset should be backed up and emptied at the same time of the dataset backup. This will eliminate the need of identifying any start point for the forward recovery process, as is done by specifying any of the 'FROM' keywords on the restore command. There may be a need however to use the 'TO' keywords to identify an end point.

For example, let's say that there are seven nightly jobs that update a file, called A, B, C, D, E, F, and G. The situation we are setting up a recovery for is a media failure that occurred while job F was running for the third time this week. The job stream can be restarted with job E. We need to restore up to and including job D that ran on the third day. So, the RESTORE command would be:

```
RESTORE   FORWARD,TOJOB=D,TODATE=day3
```

This would apply the updates for all seven jobs on the first two days, and up through and including job D on the third day. When completed, the dataset is ready for rerunning job E, and moving forward.

47.01 CONTINUED . . .

BACKOUT RECOVERY PROCESS

The backout recovery is a process where the most recent updates to a dataset are removed. This is accomplished by updating the dataset using the before images of records, which are the images of records prior to being updated or deleted. Before images will also include images of inserted records, which will cause a record deletion by a backout recovery. A backout recovery is useful for restoring a dataset to the contents it had prior to the start of job or job steps that updated the dataset. While a forward recovery could be used if all of the after images have been collected, it will probably be faster just to backout the updates from the failed job rather than reapply what might be several days worth of updates.

To be able to perform a backout recovery process, the journal must contain the before images of the updated records. This is accomplished by specifying either JRNAD=BOTH or JRNAD=BEFORE on either a CREATE or ACCESS override. If you wanted to always be able to backout the updates from any job, then be sure to provide the specification on a CREATE override, just as you would with when using a forward recovery. However, if you only wanted to provide backout ability for selected jobs, the ACCESS override could be specified for the jobs on which you wanted backout ability.

When setting up the control card for a backout recovery, generally you will want to backout the updates from one or more jobs and /or job steps that are at the end of the journal dataset. So, you will not need to specify any of the ‘TO’ keywords to terminate selection, but you would specify some of the ‘FROM’ keywords, or perhaps just the JOBNAME and/or STEPNAME keywords. For example, if job G was the last update job to run, and it abended in one of the steps updating the IAM file. To backout all of the updates made by job G, the restore command would be:

```
RESTORE BACKOUT,JOBNAME=G
```

If the journal dataset may contain multiple jobs by the name of G, the control statement would be changed to:

```
RESTORE BACKOUT,JOBNAME=G,FROMDATE=today,FROMTIME=lasttime
```

Another example would be if you have run jobs A, B, C, D, E, F, and G, with job G abending. For logistical reasons you need to restart with job E. The control statement would look like:

```
RESTORE BACKOUT,FROMJOB=E,FROMDATE=today
```

COMMAND SUMMARY

The IAMJREST program has the following commands / functions:

RESTORE: – This will invoke either the forward or backout recovery processing.

Please note that IAMJREST will only perform one command per execution.

47.02 IAMJREST - JCL REQUIREMENTS

The JCL statements required to execute IAMJREST are as follows:

EXECUTE STATEMENT Specifies the name of the IAM journal recovery / restore program – IAMJREST. For a file restore operation, sufficient storage on the REGION parameter must be specified to include storage for your system SORT. The SORT is invoked dynamically for sorting the selected records from the journal dataset to expedite recovery processing. A typical execute statement would be:

```
//RESTORE   EXEC PGM=IAMJREST,REGION=64M
```

DD STATEMENTS The following table identifies the DD statements required by IAMJREST.

DD Name	Description
STEPLIB or JOBLIB	An optional DD statement that specifies the library containing the IAM program load modules. This DD is not necessary if IAM is in the Link List, as is recommended.
SYSPRINT	Required DD statement that specifies where the IAMJREST messages are to be printed. This is usually a SYSOUT dataset.
SYSOUT	Required DD statement for RESTORE command that specifies the output message dataset for the SORT. This is usually a SYSOUT dataset.
IAMINFO	Optional DD statement that specifies the output report dataset that contains the statistics for the IAM file activity that is performed by the IAMJREST program. This is usually a SYSOUT dataset.
IAMJRNL	Optional DD statement that specifies the IAM journal dataset(s) for the IAM file to be recovered. If the journal dataset is backed up daily to a sequential file, the backup copies can be concatenated with the current journal dataset. If such concatenation is done, be sure it is done such that the records will be read by IAMJREST in ascending time sequence. The oldest journal data must be first in the concatenation, and the most recent must be the last in the concatenation. If not specified, then IAMJREST will default to using the current IAM RLS log datasets to attempt to perform the requested recovery.
IAMFILE	Optional DD statement for RESTORE command from non-RLS journal that specifies the IAM file to be restored by IAMJREST. This DD is ignored for RESTORE commands from RLS journals.
SORTWKnn SORTLIB	These DD statements may be required when performing a RESTORE operation to utilize the system SORT. Be sure to provide adequate SORT work space (as specified on the SORTWKnn DD statements.)
SYSIN	Required DD statement that contains the card image input of the command to be executed by IAMJREST. This is normally a sysin (DD *) type of dataset.

47.03 IAMJREST - RESTORE COMMAND

RESTORE COMMAND STATEMENT

The RESTORE command will perform the indicated file recovery process. Optional record selection capability is provided based on the additional keywords. If processing a journal from non-RLS processint, the RESTORE command will update the IAM dataset specified on the IAMFILE DD card with the data record images from the IAM Journal dataset specified on the IAMJRNL DD card. If processing from an IAM RLS journal, the RESTORE command will update all IAM datasets that meet the specified criteria, unless limited by the DSN keyword. Please note that updates performed by IAMJREST will not be written to the journal by the IAM journaling function.

When multiple selection criteria are specified, all of the indicated conditions must be met for the journal record to be selected for the recovery process.

RESTORE	
FORWARD | BACKOUT	
[,DSN=IAM dataset name**]**	
[,FROMDATE=yyyyddd **]**	**[,FROMJOB=**jobname **]**
[,FROMSTEP=stepname **]**	**[,FROMTIME=**hhmmss **]**
[JOBDATE=yyyydddd**]**	**[,JOBID=**jes job id**]**
[,JOBNAME=jobname **]**	**[,JOBTIME=**hhmmss**]**
[LOGSTREAMNAME=]	**[,SEQUENTIAL]**
[,STEPNAME=stepname **]**	**[SYNCPOINT]**
[,TODATE=yyyyddd **]**	**[,TOJOB=**jobname **]**
[,TOSTEP=stepname **]**	**[,TOTIME=**hhmmss **]**
[,TRANID=transaction name **]**	**[,TRANNO=**nnnnn**]**

Figure 96: IAMJREST RESTORE Command Operands

RESTORE COMMAND OPERANDS

Operand	Description
FORWARD	Specifies that a forward recovery will be performed. The forward recovery process is where the inserts, updates, and deletes from the selected journal records are performed on the IAM dataset. A forward recovery is done after restoring an IAM dataset, providing a mechanism to bring the file up to a specified point in time from the journal. To perform a forward recovery, the after images must be captured by the IAM journal, as indicated by specifying either JRNAD=BOTH or JRNAD=AFTER. Default is FORWARD recovery.
BACKOUT	Specifies that a backout (or backwards) recovery will be performed. The backout recovery process is where any updates, inserts, or deletes performed from the selected journal records will be removed from the file by updating the dataset with the version of the record prior to the update. Records that were inserted will be deleted, and records that were deleted will be inserted. This process is accomplished by sorting the selected journal before images into a descending time sequence. To perform a backout recovery, the IAM journal must have before images, as indicated by specifying either JRNAD=BOTH or JRNAD=BEFORE. Default is FORWARD recovery.

47.03 CONTINUED . . .

DSN= Specifies the IAM dataset for which recovery is to be performed.

Default for non-RLS journal file input is the dataset identified by the IAMFILE DD statement. For RLS processing, the default is all IAM files processed for the selected journal records.

FROMDATE= Specifies the lower time limit for journal records to be included in the recovery process. All preceding records are ignored. Must be specified in the form of 'yyyyddd', where yyyy is the 4-digit year value, and ddd is the 3-digit julian day value.

Default is none, the date is not used for starting journal record selection.

FROMJOB= A 1 to 8 character value for jobname that specifies that journal records will be selected for processing starting with the first journal record found for the specified jobname. All preceding journal records are ignored.

Default is none, the jobname is not used for starting journal record selection.

FROMSTEP= A 1 to 8 character value for step name that specifies that journal records will be selected for processing starting with the first journal record found for the specified step name. All preceding journal records are ignored.

Default is none, the step name is not used for starting journal record selection.

FROMTIME= Specifies that journal records will be selected for processing starting with the first journal record found for the specified time. All preceding journal records are ignored. The time is specified as either a 4 character 'hhmm' or a 6 character 'hhmmss' value, using an 'hh' value based on a 24-hour clock.

Default is none, the time will not be used for starting journal record selection.

JOBDATE= For a backout recovery from an IAM RLS journal, specifies the start date of the job from where recovery is to start. Additionally, when specified for an execution where IAMJREST is selecting from the IAM RLS journal datasets, IAMJREST will find the journal that will have the data for this date.

Default is no date criteria will be used, and when automatic selection of the IAM RLS journal is being done, IAMJREST will use the currently active IAM RLS journal.

JOBID= For a backout recovery from the IAM RLS journal datasets, specifies the JES JOBID in the format of JOB##### or J####### as applicable for your installation.

Default is that the JOBID is not used for selection. For a backout recovery from the IAM RLS journal, either a JOBNAME and / or JOBID must be specified.

JOBNAME= A 1 to 8 character value specifying that only those journal records with a matching jobname will be selected for the recovery process.

A special value of * can be used, which indicates the jobname of the job executing the IAMJREST recovery.

Default is none, there is no jobname restriction on journal record selection.

JOBTIME= For a backout recovery from the IAM RLS journals, specifies the starting time of the job for which the backout is to occur. This value will also be used by IAMJREST when selecting which IAM RLS journal to use.

Default is that job start time is not used for selection.

47.03 CONTINUED . . .

LOGSTREAMNAME= Optional keyword that indicates that IAMJREST will use the specified MVS system logger stream for the recovery process. The LOGSTREAMNAME is only applicable for IAM RLS journals.

Default is that the standard sequential log datasets will be used.

SEQUENTIAL Optional keyword that indicates that during a FORWARD recovery, all updated images will be applied as they occur in the journal.

Default is that only the last image of a particular record as identified by key or RBA will be used to update the file to save processing time.

STEPNAME= A 1 to 8 character value specifying that only those journal records with a matching step name will be selected for the recovery process.

Default is none, there is no step name restriction on journal record selection.

TODATE= Specifies the journal records up to and including the specified date will be eligible for the recovery process. Must be specified in the form of ‘yyyyddd’, where yyyy is the 4-digit year value, and ddd is the 3-digit julian day value.

Default is none, there is no date limitation on journal record selection.

TOJOB= A 1 to 8 character value specifying that journal records up to and including the specified job will be eligible for the recovery process. All journal records after those for the specified job are ignored.

Default is none, there is no jobname value for terminating journal record selection.

TOSTEP= A 1 to 8 character value specifying that journal records up to and including the specified step will be eligible for the recovery process. All journal records after those for the specified step are ignored.

Default is none, there is no step name value for terminating journal record selection.

TOTIME= A time value in the form of either a 4 character ‘hhmm’ or 6 character ‘hhmmss’ specifying that all journal records up to and including the specified time value are eligible for the recovery process. Journal records with time values higher than the specified time value will be ignored.

Default is none, there is no upper time limit for terminating journal record selection.

47.04 IAMJREST EXAMPLES

A few examples of running IAMJREST are shown below. Additional examples, along with information on setting up and using the IAM Journaling facility are provided in Section 10-88 IAM Journal and Recovery.

EXAMPLE A: FORWARD RESTORE EXAMPLE

This example will show all the JCL and the control card to run the forward recovery that was discussed under Forward Recovery in section 48.01. For this example, we are recovering up to and including all the updates done by 'jobD' that was run on July 8,1998 (julian 1998.189), that had ended by 11:15pm.

```
//FORWARD   EXEC PGM=IAMJREST,REGION=64M
//SYSPRINT   DD  SYSOUT=*
//SYSOUT     DD  SYSOUT=*
//IAMINFO    DD  SYSOUT=*
//IAMJRNL    DD  DSN=my.iam.ksd.log,DISP=OLD
//IAMFILE    DD  DSN=my.iam.ksd,DISP=OLD
//SORTWK01   DD  UNIT=SYSDA,SPACE=(CYL,(50,10))
//SORTWK02   DD  UNIT=SYSDA,SPACE=(CYL,(50,10))
//SORTWK03   DD  UNIT=SYSDA,SPACE=(CYL,(50,10))
//SYSIN      DD  *
  RESTORE FORWARD,TOJOB=jobD,TODATE=1998189,TOTIME=2315
/*
```

Figure 97: Example of Forward Recovery with TOJOB, TODATE, and TOTIME. (EX4704A)

EXAMPLE B: BACKOUT RESTORE

This example will show all the JCL and the control card to run a backout recovery. The circumstance is that we want to backout the updates done by job UPDG, which has been run several times. We just want to backout the updates from the last run, which started at 10:30am.

```
//BACKOUT   EXEC PGM=IAMJREST,REGION=64M
//SYSPRINT   DD  SYSOUT=*
//SYSOUT     DD  SYSOUT=*
//IAMINFO    DD  SYSOUT=*
//IAMJRNL    DD  DSN=my.iam.ksd.log,DISP=OLD
//IAMFILE    DD  DSN=my.iam.ksd,DISP=OLD
//SORTWK01   DD  UNIT=SYSDA,SPACE=(CYL,(50,10))
//SORTWK02   DD  UNIT=SYSDA,SPACE=(CYL,(50,10))
//SORTWK03   DD  UNIT=SYSDA,SPACE=(CYL,(50,10))
//SYSIN      DD  *
      RESTORE BACKOUT,FROMJOB=UPDG,FROMDATE=1998189,FROMTIME=1030
/*
```

Figure 98: Example of a BACKOUT Restore (EX4704B)

EXAMPLE C: IAM RLS BACKOUT RESTORE

This example will show the JCL to run a backout recovery from within an abending job that was accessing IAM datasets with IAM RLS. To automate the recovery, the IAMJREST job step will be invoked only if a preceding job step has abended. The backout will be performed from the last IAM batch SYNCPOINT that was performed prior to the abend. This recovery will use the current IAM RLS log dataset(s).

```
//BACKOUT   EXEC PGM=IAMJREST,COND=ONLY,REGION=0M
//SYSPRINT   DD  SYSOUT=*
//SYSOUT     DD  SYSOUT=*
//IAMINFO    DD  SYSOUT=*
//SYSIN      DD  *
RESTORE BACKOUT,SYNCPOINT,JOBNAME=*
/*
```

Figure 99: Example of an IAM RLS Backout Restore (EX4704C)

48.01 IAMJUTIL - PROGRAM OVERVIEW

IAMJUTIL is a utility program that provides functions to assist in managing the IAM journals. These capabilities include:

- A **COPY** function that will copy all or selected records from the journal file to another data set. This can be used to offload or backup the journal file(s).
- A **DUMP** function that can be used to print off all or selected records from an IAM journal.
- The **MVSLOG** function to remove old data from the MVS System Logger being used by IAM RLS.
- A **SCAN** function, that will produce a summary report of the data contained on a journal file, to indicate what job(s) and step(s) have data in the journal.

48.02 IAMJUTIL - JCL REQUIREMENTS

The JCL statements required to execute IAMJUTIL are as follows:

EXEC STATEMENT Specifies the IAM Journal Utilities program name – IAMJUTIL.

DD STATEMENTS The following table describes the required DD statements for running IAMJUTIL.

DD Name	Description
STEPLIB or JOBLIB	Specifies the IAM Load Module Library. This DD statement may be omitted if the IAM load library is in the link list.
SYSPRINT	Required DD statement that specifies where the printed output is to go, normally it is a SYSOUT=* dataset.
SYSIN	Required DD statement that specifies the control card input dataset, normally a DD * dataset.
IAMJRNL	Required DD statement that specifies the input IAM journal datasets.
JRNLOUT	Specifies the output journal file when a COPY operation is being performed. This is a sequential output disk or tape file, with DCB characteristics of the input journal dataset.

48.03 IAMJUTIL - COPY COMMAND

The COPY command is used to copy selected IAM Journal file records to another file for backup or other purposes. The selection operands act in a logical AND manner. A journal record has to meet all of the selection criteria to be eligible to be copied. To determine the contents of the IAM journal file, use the SCAN command.

COPY	
[AFTER]	**[,BEFORE]**
[,DATAONLY]	**[,DSN=**dsn**]**
[,FROMBLOCKID=9999999999999999**]**	**[,FROMBLOCKNO=**999999**]**
[,FROMDATE=yyyyddd**]**	**[,FROMTIME=**hhmmssth**]**
[,JLOG]	**[,JOBID=**cccccccc**]**
[,JOBNAME=cccccccc**]**	**[,LOGSTREAMNAME=**streamname**]**
[,OUTDD=cccccccc**]**	**[,REUSE]**
[,STEPNAME=cccccccc**]**	**[,TOBLOCKID=**9999999999999999**]**
[,TOBLOCKNO=999999**]**	**[,TODATE=**yyyyddd**]**
[,TOTIME=hhmmssth**]**	**[,TRANID=**cccc**]**
[,TRANNO=99999**]**	

Figure 100: IAMJUTIL COPY Command Operands

COPY OPERANDS The following operands can be specified on the COPY command. The underscored portion indicates the minimum abbreviation that can be specified for the keyword.

Operand	Description
AFTER	Optional keyword that specifies IAMJUTIL is to copy only the journal records containing after images. Suggested use is when offloading an IAM RLS journal from which there will be no future backout recoveries performed, copying only the after images will reduce the size of the offloaded dataset. After images are used for forward recoveries. Default is that both before and after images are eligible to be copied.
BEFORE	Optional keyword that specifies IAMJUTIL is to copy only the journal records containing before images. Before images are only used for backout recoveries. Default is that both before and after images are eligible to be copied.
DATAONLY	Optional keyword that specifies IAMJUTIL is to copy only the IAM data records to the output file. The journal record heading information will be eliminated. Default is that the entire journal record will be copied, including the descriptive header information.
DSN=	Optional keyword that specifies IAMJUTIL is to copy from an IAM RLS journal only those journal records for the specified IAM dataset. Default is that the journal records for all of the IAM data sets will be copied.
FROMBLOCKID=	Optional keyword that specifies IAMJUTIL is to copy journal records from the specified MVS system logger dataset starting with the specified block id number. Default is to start with the first record in the MVS system logger.

48.03 CONTINUED . . .

FROMBLOCKNO= Optional keyword that specifies IAMJTUIL is to copy journal records from a sequential IAM journal file starting from the specified relative block number.

Default is to start with the first block in the input journal file.

FROMDATE= Optional keyword that specifies IAMJTIL will copy records starting with those that have the specified date. The date is specified as a julian date in the format yyyyddd.

Default is that the copy will begin with the first record in the journal file, regardless of the date.

FROMTIME= Optional keyword that specifies IAMJUTIL will copy records starting with those that have the specified time. The format of the time, which is based on 24-hour period, is hhmmssth.

Default is that the copy will begin with the first record in the journal file, regardless of the time.

JLOG Optional keyword that specifies IAMJUTIL will produce a printed log of all activity performed by the copy command.

Default is no detailed log will be produced.

JOBID= Optional keyword that specifies IAMJUTIL will copy records for the JES JOBID (usually JOBnnnnn or Jnnnnnnn) specified.

Default is that records for all jobs are eligible to be copied.

JOBNAME= Optional keyword that specifies IAMJUTIL will copy only the journal records that have a matching jobname. If multiple jobs with the same name are encountered, then all of the jobs with a matching name will be copied, unless restricted by other operands.

Default is that journal records for all jobs are eligible to be copied.

LOGSTREAMNAME= Specifies the name of the MVS system logger dataset from which IAMJUTIL will copy records. This keyword is required when processing an IAM RLS system logger journal dataset.

Default is that IAMJUTIL assumes the copy is from a sequential journal dataset.

OUTDD= Optional keyword that specifies the name of the DD statement that defines the dataset into which the data will be copied.

Default is JRNLOUT.

REUSE Optional keyword that can be used when copying data from an IAM RLS journal file. This will set an indication to IAM RLS that the journal file can be reused when needed, but in the interim period the data in the journal remains available for automated backout processing. This is recommended when offloading an active IAM RLS journal.

Default is that IAMJUTIL will not set the reusability indicator for this journal dataset.

STEPNAME= Optional keyword that specifies that IAMJUTIL will copy only those records with the specified stepname.

Default is that journal records are eligible to be copied regardless of the stename.

TOBLOCKID= Optional keyword that can be used when copying journal data from an MVS system logger dataset, that indicates IAMJUTIL will copy only data up to and including the specified system logger record ID.

Default is that copying will end with the last record that is currently in the MVS system logger dataset.

48.03 CONTINUED . . .

TOBLOCKNO= Optional keyword when copying from a sequential IAM journal, that specifies IAMJUTIL is to copy all data up that is contained in the blocks up to and including the specified relative block number.

Default is that copying will end with the last record that is in the sequential journal dataset.

TODATE= Optional keyword that specifies IAMJUTIL is to copy all journal records up to and including the specified date. The date is specified as a julian date in the format yyyyddd.

Default is that copying will end with the last record found in the input journal dataset(s).

TOTIME= Optional keyword that specifies IAMJUTIL is to copy all journal records up to and including the specified time. The time is specified in the format of hhmmssth.

Default is that copying will end with the last record found in the input journal dataset(s).

TRANID= Optional keyword that specifies IAMJUTIL will copy only those records from an IAM RLS journal for the specified CICS transaction name. TRANID specifies the 1 to 4 character transaction name.

Default is that all transactions are eligible to be copied.

TRANNO= Optional keyword that specifies IAMJUTIL will copy only those records with the specified CICS transaction number. TRANNO specifies a 1 to5 digit transaction number.

Default is that all transactions are eligible to be copied.

EXAMPLE A: USING IAMJUTIL COPY TO OFFLOAD DATA FROM AN IAM RLS JOURNAL

This is an example of how to use IAMJUTIL to offload data from an IAM RLS journal into an accumulation dataset, and mark the IAM RLS journal as being ready for reuse. Only the after images are being copied, because the journal data will be used only to perform a forward recovery from a restored datasets. Backout recoveries will not be possible from the accumulation dataset. Backout recoveries can still be performed from the IAM RLS journal until it is actually reused by IAM RLS.

```
//EX4803A  EXEC PGM=IAMJUTIL
//SYSUDUMP  DD  SYSOUT=*
//SYSPRINT  DD  SYSOUT=*
//SYSOUT    DD  SYSOUT=*
//IAMJRNL   DD  DISP=SHR,DSN=PROD.IAMRLS.JOURNAL.LOGDSN2
//JRNLOUT   DD  DISP=MOD,DSN=PROD.IAMRLS.OFFLOAD.JOURNAL
//SYSIN     DD  *
 COPY AFTER,OUTDD=JRNLOUT,REUSE
/*
```

Figure 101: Example of Offloading data from an IAM RLS Journal (EX4803A)

48.04 IAMJUTIL - DUMP COMMAND

The DUMP command is used to produce a printed dump of selected IAM Journal file records. This may be useful if there are any problems when performing RESTORE functions with IAMJREST. The selection operands act in a logical AND manner. A journal record has to meet all of the selection criteria to be eligible to be copied. To determine what jobs and steps have written any journal records, use the IAMJUTIL SCAN command.

DUMP	
[ALL]	**[,BLOCKS]**
[,COUNT=999999**]**	**[,DSN=**dsn**]**
[,FROMBLOCKID=9999999999999999**]**	**[,FROMBLOCKNO=**999999**]**
[,FROMDATE=yyyyddd**]**	**[,FROMTIME=**hhmmssth**]**
[,HEADERONLY]	**[,JLOG]**
[,JOBID=cccccccc**]**	**[,JOBNAME=**cccccccc**]**
[,LOGSTREAMNAME=streamname**]**	**[,STEPNAME=**cccccccc**]**
[,TOBLOCKID=9999999999999999**]**	**[,TOBLOCKNO=**999999**]**
[,TODATE=yyyyddd**]**	**[,TOTIME=**hhmmssth**]**
[,TRANID=cccc**]**	**[,TRANNO=**99999**]**

Figure 102: IAMJUTIL DUMP Command Operands

DUMP OPERANDS The following operands can be specified on the DUMP command. The underscored portion indicates the minimum abbreviation that can be specified for the keyword.

Operand	Description
ALL	Optional keyword that specifies IAMJUTIL is to dump the entire contents of each journal record, including the header information and the actual data record. Default is that only the data portion is printed.
BLOCKS	Optional keyword that specifies IAMJUTIL is to dump the journal dataset by block, not by records. Default is to dump each journal record.
COUNT=	Optional keyword that specifies IAMJUTIL is to dump only the specified number of records. Default is that all journal records are eligible.
DSN=	Optional keyword that specifies IAMJUTIL is to dump from an IAM RLS journal only those journal records for the specified IAM dataset. Default is that the journal records for all of the IAM data sets will be dumped.
FROMBLOCKID=	Optional keyword that specifies IAMJUTIL is to dump journal records from the specified MVS system logger dataset starting with the specified block id number. Default is to start with the first record in the MVS system logger.
FROMBLOCKNO=	Optional keyword that specifies IAMJTUIL is to dump the journal records from a sequential IAM journal file starting from the specified relative block number. Default is to start with the first block in the input journal file.

48.04 CONTINUED . . .

FROMDATE= Optional keyword that specifies IAMJTIL will dump records starting with those that have the specified date. The date is specified as a julian date in the format yyyyddd.

Default is that the dump will begin with the first record in the journal file, regardless of the date.

FROMTIME= Optional keyword that specifies IAMJUTIL will dump records starting with those that have the specified time. The format of the time, which is based on 24-hour period, is hhmmssth.

Default is that the dump will begin with the first record in the journal file, regardless of the time.

HEADERONLY Optional keyword that specifies IAMJUTIL will dump only the header portion of each journal record.

Default is that IAMJUTIL will dump only the data portion of each journal record.

JLOG Optional keyword that specifies IAMJUTIL will produce a printed log of all activity performed by the dump command.

Default is no detailed log will be produced.

JOBID= Optional keyword that specifies IAMJUTIL will dump records for the JES JOBID (usually JOBnnnnn or Jnnnnnnn) specified.

Default is that records for all jobs are eligible to be dumped.

JOBNAME= Optional keyword that specifies IAMJUTIL will dump only the journal records that have a matching jobname. If multiple jobs with the same name are encountered, then all of the jobs with a matching name will be dumped, unless restricted by other operands.

Default is that journal records for all jobs are eligible to be dumped.

<u>LOGSTREAM</u>NAME= Specifies the name of the MVS system logger dataset from which IAMJUTIL will dump records. This keyword is required when processing an IAM RLS system logger journal dataset.

Default is that IAMJUTIL assumes the dump is from a sequential journal dataset.

<u>STEP</u>NAME= Optional keyword that specifies that IAMJUTIL will dump only those records with the specified stepname.

Default is that journal records are eligible to be dumped regardless of the stename.

<u>TOBLOCKID=</u> Optional keyword that can be used when dumping journal data from an MVS system logger dataset, that indicates IAMJUTIL will dump only data up to and including the specified system logger record ID.

Default is that dumping will end with the last record that is currently in the MVS system logger dataset.

<u>TOBLOCKNO=</u> Optional keyword when dumping from a sequential IAM journal, that specifies IAMJUTIL is to dump all data up that is contained in the blocks up to and including the specified relative block number.

Default is that dumping will end with the last record that is in the sequential journal dataset.

48.04 CONTINUED . . .

TODATE= Optional keyword that specifies IAMJUTIL is to dump all journal records up to and including the specified date. The date is specified as a julian date in the format yyyyddd.

Default is that dumping will end with the last record found in the input journal dataset(s).

TOTIME= Optional keyword that specifies IAMJUTIL is to dump all journal records up to and including the specified time. The time is specified in the format of hhmmssth.

Default is that dumping will end with the last record found in the input journal dataset(s).

TRANID= Optional keyword that specifies IAMJUTIL will dump only those records from an IAM RLS journal for the specified CICS transaction name. TRANID specifies the 1 to 4 character transaction name.

Default is that all transactions are eligible to be dumped.

TRANNO= Optional keyword that specifies IAMJUTIL will dump only those records with the specified CICS transaction number. TRANNO specifies a 1 to5 digit transaction number.

Default is that all transactions are eligible to be dumped.

EXAMPLE A: PRINTING (DUMPING) SELECTED JOURNAL RECORDS

In this example of the DUMP command, records for the specified job step will be printed.

```
//EX4804A   EXEC PGM=IAMJUTIL
//SYSUDUMP   DD   SYSOUT=*
//SYSPRINT   DD   SYSOUT=*
//IAMJRNL    DD   DISP=SHR,DSN=GFM.LOGDSN2.JOURNAL
//SYSOUT     DD   SYSOUT=*
//SYSIN      DD   *
  DUMP ALL,JOB=jobname,STEP=stepname,JOBID=JOBnnnnn
/*
```

Figure 103: Example of Dumping Selected Records (EX4804A)

48.05 IAMJUTIL - MVSLOG COMMAND

The IAMJUTIL MVSLOG command is provided to help administer the MVS logstream, if users have specified the use of the logstream by IAM RLS. This command provides a mechanism to remove old records from the logstream, providing such is not done automatically or by other means.

MVSLOG	**DELETE**
[,ALL]	**[,BLOCKID=**nnnnnnn**]**
[,DATE=yyyyddd**]**	**[,LOGSTREAMNAME=**logdsn**]**
[,TIME=hhmmssth**]**	

Figure 104: IAMJUTIL MVSLOG Operands

MVSLOG OPERANDS The following describes the operands that can be specified on the MVSLOG command. When an abbreviation is allowed, the underscored portion of the keyword represents the minimum allowable abbreviation. In addition to the required operand of DELETE, additional operands must be specified to indicate what data is to be removed from the log stream.

Operand	Description
DELETE	Required keyword that specifies IAMJUTIL is to perform a record deletion request on the specified MVS log stream.
ALL	Optional keyword that specifies that IAMJUTIL is to request the deletion of all of the records in the specified log stream. Default is none, either ALL, BLOCKID or DATE must be specified.
BLOCKID=	Optional keyword that specifies that IAMJUTIL is to request deletion of all records preceding the specified block id. Default is none, either ALL, BLOCKID or DATE must be specified.
DATE=yyyyddd	Optional keyword that specifies that IAMJUTIL is to request deletion of all records preceding the specified date, which must be specified as a julian date. Default is none, either ALL, BLOCKID or DATE must be specified.
LOGSTREAMNAME=	Required keyword that specifies the name of the MVS System Logger log stream that was used by IAM RLS.
TIME=hhmmssth	Option keyword that can be specified with the DATE operand, that will specify to IAMJUTIL to request deletion of all records in the log stream preceding the specified date and time. Default is 00000000.

48.05 CONTINUED . . .

EXAMPLE A: DELETING OLD RECORDS FROM LOG STREAM

The example below shows the JCL and control cards necessary to delete records from an MVS Log Stream that are older than the specified date.

```
//EX4805A   EXEC PGM=IAMJUTIL
//SYSUDUMP   DD   SYSOUT=*
//SYSPRINT   DD   SYSOUT=*
//SYSOUT     DD   SYSOUT=*
//SYSIN      DD   *
 MVSLOG DELETE,DATE=2002138,
        LOGSTREAMNAME=IDP.CICS.LOGSTRM
/*
```

Figure 105: Examle of Deleting Old Records from MVS Log Stream (EX4805A)

EXAMPLE B: DELETING ALL RECORDS FROM AN MVS LOG STREAM

The example below demonstrates how to delete all data from a log stream. This might be helpful for test systems, where the journaled data does not have to be saved, or after offloading the data from the log stream.

```
//EX4805B   EXEC PGM=IAMJUTIL
//SYSUDUMP   DD   SYSOUT=*
//SYSPRINT   DD   SYSOUT=*
//SYSOUT     DD   SYSOUT=*
//SYSIN      DD   *
 MVSLOG DELETE,ALL,LOGSTREAMNAME=IDP.CICS.LOGSTRM
/*
```

Figure 106: Example of Deleting all records in a Log Stream (EX4805B)

48.46 IAMJUTIL - SCAN COMMAND

SCAN COMMAND STATEMENT

The SCAN command can be used to determine the contents of IAM journal data sets. A report is generated which describes all of the jobs and steps that have written out data to the specified input journal data set, including dates and times.

SCAN	
[DSN=dsn**]**	**[,FROMBLOCKID=**9999999999999999**]**
[,FROMBLOCKNO=999999**]**	**[,FROMDATE=**yyyyddd**]**
[,FROMTIME=hhmmssth**]**	**[,JLOG]**
[,JOBID=cccccccc**]**	**[,JOBNAME=**cccccccc**]**
[,LOGSTREAMNAME=streamname**]**	**[,STEPNAME=**cccccccc**]**
[,TOBLOCKID=9999999999999999**]**	**[,TOBLOCKNO=**999999**]**
[,TODATE=yyyyddd**]**	**[,TOTIME=**hhmmssth**]**
[,TRANID=cccc**]**	**[,TRANNO=**99999**]**

Figure 107: IAMJUTIL SCAN Command Format

SCAN COMMAND OPERANDS

Operand	Description
DSN=	Optional keyword that specifies IAMJUTIL is to scan from an IAM RLS journal only those journal records for the specified IAM dataset. Default is that the journal records for all of the IAM data sets will be scanned.
FROMBLOCKID=	Optional keyword that specifies IAMJUTIL is to scan journal records from the specified MVS system logger dataset starting with the specified block id number. Default is to start with the first record in the MVS system logger.
FROMBLOCKNO=	Optional keyword that specifies IAMJTUIL is to scan the journal records from a sequential IAM journal file starting from the specified relative block number. Default is to start with the first block in the input journal file.
FROMDATE=	Optional keyword that specifies IAMJTIL will scan records starting with those that have the specified date. The date is specified as a julian date in the format yyyyddd. Default is that the scan will begin with the first record in the journal file, regardless of the date.
FROMTIME=	Optional keyword that specifies IAMJUTIL will scan records starting with those that have the specified time. The format of the time, which is based on 24-hour period, is hhmmssth. Default is that the scan will begin with the first record in the journal file, regardless of the time.
JLOG	Optional keyword that specifies IAMJUTIL will produce a printed log of all activity performed by the scan command. Default is no detailed log will be produced.

48.46 CONTINUED . . .

JOBID= Optional keyword that specifies IAMJUTIL will scan records for the JES JOBID (usually JOBnnnnn or Jnnnnnnn) specified.

Default is that records for all jobs are eligible to be scanned.

JOBNAME= Optional keyword that specifies IAMJUTIL will scan only the journal records that have a matching jobname. If multiple jobs with the same name are encountered, then all of the jobs with a matching name will be scanned, unless restricted by other operands.

Default is that journal records for all jobs are eligible to be scanned.

LOGSTREAMNAME= Specifies the name of the MVS system logger dataset from which IAMJUTIL will scan records. This keyword is required when processing an IAM RLS system logger journal dataset.

Default is that IAMJUTIL assumes the data is from a sequential journal dataset.

STEPNAME= Optional keyword that specifies that IAMJUTIL will scan only those records with the specified stepname.

Default is that journal records are eligible to be scanned regardless of the stename.

TOBLOCKID= Optional keyword that can be used when scanning journal data from an MVS system logger dataset, that indicates IAMJUTIL will scan only data up to and including the specified system logger record ID.

Default is that scanning will end with the last record that is currently in the MVS system logger dataset.

TOBLOCKNO= Optional keyword when scanning from a sequential IAM journal, that specifies IAMJUTIL is to scan all data up that is contained in the blocks up to and including the specified relative block number.

Default is that scanning will end with the last record that is in the sequential journal dataset.

TODATE= Optional keyword that specifies IAMJUTIL is to scan all journal records up to and including the specified date. The date is specified as a julian date in the format yyyyddd.

Default is that scanning will end with the last record found in the input journal dataset(s).

TOTIME= Optional keyword that specifies IAMJUTIL is to scan all journal records up to and including the specified time. The time is specified in the format of hhmmssth.

Default is that scanning will end with the last record found in the input journal dataset(s).

TRANID= Optional keyword that specifies IAMJUTIL will scan only those records from an IAM RLS journal for the specified CICS transaction name. TRANID specifies the 1 to 4 character transaction name.

Default is that all transactions are eligible to be scanned.

TRANNO= Optional keyword that specifies IAMJUTIL will scan only those records with the specified CICS transaction number. TRANNO specifies a 1 to5 digit transaction number.

Default is that all transactions are eligible to be scanned.

48.46 CONTINUED . . .

EXAMPLE A: BASIC SCAN OF A JOURNAL

This example shows how to run a scan of the current journal data set. The report produced will contain a list of all the job/job steps that updated the corresponding IAM data set, along with time stamps of the first and last record for each job step.

```
//EX4806A   EXEC PGM=IAMJUTIL
//SYSPRINT DD   SYSOUT=*
//IAMJRNL   DD   DSN=my.iam.ksd.log,DISP=SHR
//SYSIN     DD   *
  SCAN
/*
```

Figure 108: Example of Obtaining a Summary of Journal Contents (EX4806A)

THIS PAGE WAS INTENTIONALLY LEFT BLANK

80.01 INTRODUCTION

This section describes the various printer, console, and TSO messages which may be displayed by INNOVATION's IAM product and the various ABEND codes with which it may terminate.

The general foramt of IAM messages and WTORs is as follows:

IAMnnn Message on printed report.

IAMWnn IAM WTO/R message.

IAMJnn IAM Journal Exit WTO message.

IAMLnnnn IAM RLS messages, displayed on the IAM RLS RLSLOGDD, and may also be a WTO message.

IAM UTILITY RETURN CODES

IAM utility programs pass a return code at the end of the step, unless they ABEND. A return code of zero indicates that IAM has performed all functions successfully. A return code of four (004) indicates that the testing period for a trial version of the product has expired. The production version of the product library does not contain an expiration date. Any other return code is accompanied by error messages, and indicates that some kind of error has occurred during the execution.

VSAM RETURN CODES

IAM's VSAM Interface (VIF) will pass return codes comparable to the return codes set by VSAM. The return codes are passed in the RPL for I/O requests, or in the ACB for Open or Close requests. The return codes that IAM will issue are documented in Section 80.21 and 80.22.

ABEND CODES

When using IAM through the VSAM interface, IAM abides by general VSAM rules, which is to pass back error return codes and error codes rather than abending. The exception to that rule is a possible U0184 abend that can occur for diagnostic purposes for various errors when the user has added an //IAMDEBUG DD DUMMY card to a potentially failing job step. This DD statement should only be coded at the request of IAM support personnel to obtain additional documentation for a particular problem. This is because many normally occurring errors may result in a U0184 abend when the DD card is present.

IAM abend codes range from U0100 to U0999. IAM ABENDS are preceded by an IAM error message. For the VIF interface, IAM avoids intentionally abending whenever possible, but will issue appropriate error codes.

80.02 IAM WTO MESSAGES

IAMW01 DD='*ddname*' I/O ERROR CODE=X'*decb iobecb csw cccchhhhrr*'

Reason: This message is provided by the IAM processor for Compatible format files, when a file access fails with an I/O error. The message contains the following diagnostic information:

DECB— displays the two error bytes of the BDAM DECB(+1).

Possible codes are:

8000— Block not found on track

4000— Block length was incorrect

0800— Media failure – Data or Equipment check

0400— Physical End of File marker

0200— Unidentified error

0010— Requested block not within file extents

IOBECB— displays the first four bytes of the IOB.

CSW— displays the channel status word.

cccchhhhrr—displays the cylinder, head and record head number the error was detected on. The head and record number may not be accurate. IBM's message IEA000I/IOS000I, if present, contains the real track address.

Return Codes: A Return Code of 12, X'0C', and a reason code of X'04' for a read I/O error, or X'10' for an output I/O error, is stored in the RPL. The SYNAD exit will be invoked, if so specified by the program.

Action: If an IBM IEA000I/IOS000I message appears on the JCL LOG, a hardware error has occurred. The IBM message gives the sense information (ex: data check, equipment check). Examine the error information provided in the IBM message to determine the cause of the error. The format of this information is documented in the IBM data management SRL for the operating system in use. The file in question is unreadable in its present state and must be reestablished. This may be the result of a hardware error and if possible the new file should be allocated to a different physical location or volume.

For other I/O errors, those not accompanied by a hardware failure error message, determine the cause of the error. It is recommended that an IAMRECVR DIAGNOSE function be executed on the problem file, to see if there are any problems with the file integrity. Some common reasons for these types of I/O errors include:

File has been improperly moved or restored to a device type different than it was originally loaded on. Frequently, this will fail with block not found I/O error, with a DECB error code of x'8000'.

Multivolume file has been improperly moved, or improperly cataloged. Frequently, this will result in an I/O error of block not within extent, DECB error code of x'0010'.

Storage overlay of IAM I/O control blocks. When this type of error occurs, most jobs and application programs do successfully process the IAM file, as does IAMRECVR. The failure is typically limited to one or a few jobs. A SYSUDUMP will be needed to determine the cause of this type of error.

The utility IAMRECVR may be used to recover a file that is no longer usable due to I/O or logical error conditions. This utility can be used to off load records from those portions of the file that have not been physically damaged. The user program should make a decision on the action to take when this occurs, for example to continue processing without this file or to terminate processing until this file is made available. The appropriate action will depend entirely on the application and the user program's evaluation of the diagnostic information that is returned. If further assistance is needed, contact Innovation Data Processing.

80.02 CONTINUED . . .

IAMW02 REPLY RETRY WAIT OR CANCEL FOR ENQ WAIT ON '*dsname*'

Reason: The IAM file is not available to this job because some other job is currently accessing the file. This WTOR will only be issued when the IAM Global Option VSAMTWO=YES is specified, and is only applicable to Compatible Format files. A load process is always protected against concurrent access (read or update) or from another load. The Share Options specified when the file was defined determine which types of access can concurrently share the file.

Action: The operator reply determines the course of action IAM will take. The following responses are allowed:

RETRY: IAM retries the ENQUEUE for the data set. If the file is still unavailable, the message will be re-issued.

WAIT: IAM waits for the data set to become available. Caution: The job could time out.

CANCEL: IAM fails the OPEN request with a return code indicating the file is not currently available for processing. The ACBERFLG field will be set to x'A8'.

IAMW03 DD='*ddname*' FILE FULL DUE TO INSUFFICIENT CORE

Reason: This message is displayed by IAM when it finds an insert has failed because of insufficient storage to expand the overflow index. The record that was to be added was not placed in the file. Space may still be available at other locations within the file. Consequently, subsequent inserts may or may not fail depending on where in the file they are placed. Additionally, changes in the storage available within a region may allow a subsequent GETMAIN to succeed.

This message is displayed a maximum of 10 times per execution.

Return Codes: A return code of eight (8) and a reason code of 28 (x'1C') is stored in the feedback field of the RPL. This error code indicates a VSAM file full logical error. The logical error exit, LERAD, will be invoked, if so specified in the program.

Action: This message indicates that there is an insufficient above the line storage region available for IAM to expand the overflow index size. Reorganizing the file may help to reduce the storage needed. For Compatible format files, the definition of the file's OVERFLOW parameters as well as the OCOREO% and OCOREX% parameters should be reviewed. Increasing the OCOREO% value to acquire a larger overflow index area when the file is Opened may correct the problem. The job step region size or, for CICS users, the MVS IEFUSI exit, may also need adjustment so more storage will be available for use by this particular task.

IAMW04 DD='*ddname*' OPEN ERROR – DATASET NOT AVAILABLE[, IN USE BY JOB jobn]

Reason: IAM was unable to open the specified file because it was already opened by another job. For Enhanced Format files, if IAM is able to determine which JOB or user has the file, it will be identified in the message.

Return Codes: A reason code of 168(x'A8') is stored in the ACB error flags field (ACBERFLG) and the OPEN is failed with a return code of 8.

Action: Rerun the job when the indicated job has terminated or closed the IAM file that was in use.

80.02 CONTINUED . . .

IAMW05 DD='*ddname*' OPEN ERROR – MORE THAN 2 OPENS FOR UPDATE IN TASK

Reason: This message is displayed by IAM, for Compatible format files only, when it finds a task has issued multiple Opens for update against the same dataset.

Return Codes: A reason code of x'F8' is stored in the ACB error flags and the OPEN is failed with a return code of 8.

Action: Convert the file to an Enhanced Format file, which supports multiple ACB's within the same address space. Or, change the program to insure that the same file is never opened for update more than once within the processing task without first closing it.

IAMW06 DD='*ddname*' OPEN ERROR – DD STATEMENT MISSING

Reason: This message is displayed when a task has issued an OPEN, but the 'ddname' the OPEN was issued against did not exist, or some other error in open processing has occurred.

Return Codes: Reason code of x'80' is stored in the ACB error field and the open request is failed with a return code of 8.

Action: Check for other messages that may indicate the cause of the problem. Also, check to see if the missing DD statement is not the result of a misspelling. If not, add the DD statement to the JCL.

IAMW07 DD='*ddname*' I/O ERROR CODE=X'*decb iobecb csw cccchhhhrr*'

Reason: This message is displayed for Compatible Format file when a file OPEN fails with an I/O error. The format is the same as the IAMW01 message, please refer to that message for an explanation of the error codes.

Note: This message may also occur when an attempt is made to OPEN a non-IAM or non-VSAM data set as VSAM.

Return Codes: The ACB error flag is set to a value of x'B8', and the open is failed with a return code of 8.

Action: Refer to message IAMW01 for potential causes and corrective action.

IAMW08 DD='*ddname*' OPEN ERROR – INSUFFICIENT STORAGE IN REGION

Reason: This message is displayed by IAM when a file OPEN fails because storage is not available for data buffers, the index or control information. Typically this message is issued due to insufficient storage above the 16 megabyte line (31-bit addressable storage), although it can also be issued for shortage of storage below the 16 megabyte line. Below the line storage may have become depleted due to insufficient region specified for above the line storage.

Return Codes: Reason code of x'88' is stored in the ACB error flags, and the open is failed with a return code of 8.

Action: This message can occur for any number of reasons, and be corrected in various ways depending on the base cause of the problem. Frequently the problem is that the number of records in Independent or Extended Overflow result has significantly increased, causing a high demand for virtual storage for the index to the Overflow area. That problem can most easily be resolved by reorganizing the file. Please refer to the section on Storage Tuning for complete information on IAM's storage usage, and how that usage can be controlled.

80.02 CONTINUED . . .

IAMW09 DD='*ddname*' OPEN ERROR – FILE DEFINED BUT NOT LOADED

Reason: This message is displayed by IAM when a task has issued an OPEN against an IAM file but the dataset (referenced by 'ddname') for input or update processing, but the file had never been loaded as an IAM file, or an attempted load failed.

Return Codes: Reason code of x'A0' is stored in the ACB error flags, and the open request is failed with a return code of 8.

Action: The file must be successfully loaded with data before it can be used for input or update processing.

IAMW10 DD='*ddname* ' OPEN ERROR – NOT AN IAM/VSAM FILE OR CREATE FAILED

Reason: This message is displayed by IAM when a task has issued a VSAM OPEN for the dataset (referenced by 'ddname') but the file did not contain a valid IAM file descriptor block. The dataset may be the result of a load that failed, may have a misspelled data set name, or perhaps the file was damaged.

Return Codes: Reason code of x'BC' is stored in the ACB error flags, and the Open is failed with a return code of 8.

Action: First, determine if the file being opened is supposed to be an IAM or a VSAM file. If it is supposed to be an IAM file, review the output from the job which created the file for errors.

IAMW11 DD='*ddname*' DYNCORE DISABLED DUE TO INSUFFICIENT STORAGE

Reason: This message is displayed by IAM when a user requested storage for IAM's Dynamic Table option and the virtual storage is not available. Processing continues as if the request for Dynamic Tabling had not been requested.

Return Codes: No error codes or return codes are set.

Action: The user should increase the region size or decrease the value requested for DYNCORE.

IAMW12 DD='*ddname*' DATA COMPRESS ERROR xxxxxxxx xxxxxxxxxxxxxx xxxxxxxxxxxxxx

Reason: During a file access of a data compressed IAM file, IAM decompression failed. The first four bytes are the internal RDW, the remaining data is the key.

Return Codes: A logical error code of x'2C' is set in the RPL. If present and active, the program's logical error exit (LERAD) is invoked, otherwise the request is failed with a return code of 8. If the user has placed a // IAMDEBUG DD DUMMY DD card in the JCL, then the task is abended with an abend code of U0185.

Action: Run an IAMRECVR DIAGNOSE function to validate that the file is still valid. If the DIAGNOSE did not find any errors, then the cause is most likely a storage overlay. If the application did not produce a dump, rerun the job with a //IAMDEBUG DD DUMMY DD card to get a dump. If errors were found by DIAGNOSE, then a file recovery is necessary, either with IAMRECVR or other procedures that have been established. Contact Innovation for assistance.

80.02 CONTINUED . . .

IAMW13 DD='*ddname*' FILE FULL, REORGANIZATION REQUIRED

Reason: IAM was unable to add a record into a file, or unable to accept a larger size updated record. For Compatible format files, the Independent Overflow area has been filled. For Enhanced format files, IAM was not able to obtain additional DASD space to expand the size of the file. A message indicating some type of X37 abend may precede this message.

Return Codes: A reason code of 28(x'1C') is stored in the RPL error flags field (RPLERRCD) and the PUT is failed with a return code of 8. This VSAM logical error code signifies a file full error condition. If the application has provided a logical error exit (LERAD), the exit will be invoked.

Action: Reorganize the file. It may be necessary to increase the space allocation for the file, and / or to move the file to a different volume(s). For Enhanced Format files, it may be possible to free up some DASD space that is allocated to other data sets, and retry the request without doing a file reorganization.

IAMW16 DD='*ddname*' OPEN ERROR – IAM DOES NOT CURRENTLY SUPPORT LSR

Reason: For Compatible Format files only, this message is displayed by IAM when a task has issued an open for a data set with the Local Shared Resources processing option (MACRF=LSR) specified in the ACB, and a UPAD exit active in the exit list (EXLIST) specified for the ACB. Typically, this will occur for a Compatible Format file opened by CICS, without indicating that the file is not to be in any LSR pool in the CICS FCT. The OPEN fails.

Return Codes: Reason code of x'E4' is stored in the ACB error flag field and the open is failed with a return code of 8.

Action: Either convert the file to an IAM Enhanced Format file, which will support the LSR application, or if the error occurred under CICS, change the CICS FCT table to specify LSRPOOL=NONE.

IAMW17 DD='*ddname*' ADD FAILED – MORE THAN 1 ACB CONCURRENTLY UPDATING FILE.

Reason: IAM does not support concurrent file update. This message will be displayed when IAM detects an inconsistency between an Overflow block, and the index entries for that block in virtual storage. Generally such inconsistencies should only occur if another job had updated the data set since it was last OPENed by the failing application. IAM expects proper procedures to be used to prevent concurrent update as it will eventually result in damage to the file. This message is displayed a maximum of 10 times per execution.

Return Codes: A return code of eight (8) and a reason code of x'1C' (file full) is stored in the feedback field of the RPL. The logical error exit (LERAD) will be invoked, if so specified in the program.

Action: In a multi-processor environment IAM files must be protected, by a global enqueue facility or manual scheduling, from concurrent update by tasks running on different processors. Within a single processor multiple tasks will be prevented from concurrently opening a file for update by the proper use of SHAREOPTIONS. IAM defaults to a cross region SHAREOPTION 1, multiple read access users or one update user. The use of SHAREOPTIONS 3 and 4 disable IAM's enqueue protection for a file as does the IAMOVRID parameter UPDATENQ=NONE. Use these facilities with great care. Within a single task for Compatible Format files, use of multiple ACB's against an IAM file will defeat IAM's enqueue protection for that file. To preserve the integrity of your IAM files any application program that accesses an IAM Compatible Format file with multiple ACB's must be changed to ensure only one ACB is used to access the file for update, or convert the file to the IAM Enhanced File format. (For further assistance contact INNOVATION).

80.02 CONTINUED . . .

IAMW18 DD='*ddname*' USER NOT AUTHORIZED FOR UPDATE

Reason: For Enhanced Format files, IAM issued a RACROUTE to verify that the user had authority to access the data set for the desired mode, and received a return code indicating that the user is not authorized for the requested function.

Return Codes: A reason code of x'98' is stored in the ACB error flags field, and the open request is failed with a return code of 8.

Action: Correct the error and rerun the job. Error could be caused by an incorrect data set name on the DD card, or this most likely is a legitimate error.

IAMW20 DD='*ddname*' OPEN ERROR – error description

Reason: The file load process has detected an invalid, unsupported, or inconsistent parameter for the indicated file attribute. The particular attribute that is invalid is indicated in the error description text. The possibilities are:

- LRECL Specification – Must be at least as long as key offset (RKP) plus the key length, and no larger than 32,755.
- Key Length Specification – Must be greater than 1, and less than 250.
- Blocking Factor Specification – Must be between 1 and 15.
- Blocksize Specification – Must exceed the maximum record length by at least five bytes, and be at least 300 bytes.
- RKP Specification – Must be less than 4092 bytes.
- Independent Overflow Specification – For compatible format files, can not be a value that will cause the Overflow area to exceed 64,000 blocks.
- Integrated Overflow Specification – Must be between 0 and 99.
- Prime Extension Specification – For Compatible Format files, can not exceed 32,767.
- Delete Processing Request – Should not occur with the IAM / VSAM Interface.
- Data Control Area (DCA) Validation – Should not occur with the IAM / VSAM Interface.
- Blocksize Calculation – The IAM internal blocksize calculation resulted in an value that can not be used on the device that contains the IAM file.

Return Codes: A reason code of 192(X'C0') is stored in the ACB error flags field (ACBERFLG) and the open is failed with a return code of 8.

Action: Correct the invalid specification, and rerun. This error may occur when using VIF if the IAM file was allocated through some means other than an IDCAMS DEFINE and the user failed to correctly provide file specifications through a CREATE Override Control Statement. Additionally, conflicts between the file itself and the user program's internal file definition must be resolved.

80.02 CONTINUED . . .

IAMW21 DD='*ddname*' LOAD ERROR – EXCEEDED 255 EXTENTS

Reason: During the loading of a multivolume file, the file required more than 255 extents, which is more than the number of extents allowable.

Return Codes: If failure occurred on a WRITE or PUT request, the RPL is failed with a Logical Error code of x'1C', file full logical error, and the request will receive a return code of 8. If the error occurred during CLOSE, then the CLOSE will fail with a return code of 4, and the ACB error flags set to x'90'. The file is left marked in an unloaded state.

Action: Delete and redefine the data set, increasing the primary and / or secondary space allocation values so that the file will fit within 255 extents.

IAMW22 DD='*ddname*' FILE REORGANIZATION RECOMMENDED –

Reason: A file reorganization is being recommended for one of the following reasons, as indicated in the message:

1. OVERFLOW INDEX EXCEEDS 16 MEG
2. OVERFLOW EXCEEDS 1000 CYLINDERS
3. EXCEEDED 13 EXTENTS ON SINGLE VOLUME
4. EXCEEDED REQUESTED OVERFLOW RECORDS

Return Codes: None.

Action: For reasons 1 and 2, performance on the indicated file may be adversely affected due to the size of the overflow area. In particular, it may take several minutes to open the file, and sequential processing may be detrimentally affected. It is therefore recommended that the file be reorganized at the earliest convenient time to prevent further performance deterioration. The file can be quickly reorganized with FDRREORG®, or if that product is not available, then use IDCAMS REPRO.

For reason 3, because IAM files have a non-VSAM file structure, they are limited to 16 extents per volume. The file indicated currently has 14 or more extents, so future growth will be restricted. To prevent an out of space condition, action should be taken at the earliest possible time. If there is sufficient space for the file to expand on the volume that it currently resides, either use COMPAKTOR to merge extents, or reorganize the file doing a DELETE and DEFINE of the data set, specifying a larger space allocation. The current space allocation values can be determined by performing a Listcat All on the data set. If the current volume has insufficient space, and you are unable to free up sufficient space, then the data set should be moved to a different volume, where more DASD space is available.

For reason 4, the file was defined with an Overflow override, and the file has reached or exceeded that number of records in extended overflow. Presumably, the override was provided to indicate when the file should be reorganized. IAM will continue to add records to extended overflow, provided that IAM is able to obtain sufficient DASD space to do so.

IAMW23 DD='*ddname*' OPEN ERROR – FILE WAS RELOADED SINCE READ ONLY ACB OPENED

Reason: An attempt was made to open a file with a second ACB (and possibly second DD card) within the same address space for UPDATE processing. The file had previously only been eligible for read only processing. On the open for update, it was determined that the file created time stamp is different from when the file had originally been opened by the READ only ACB. This would indicate that the file has been reloaded, and the index structure can not be updated.

Return Codes: A reason code of 240(x'F0') is stored in the ACB error flags field (ACBERFLG) and the open request fails with a return code of 8.

Action: Close all other ACB's open to that data set, and then reopen.

80.02 CONTINUED . . .

IAMW24 DD='*ddname*' OPEN ERROR – PROGRAM CHECK OCCURRED DURING OPEN

Reason: While IAM was opening a file, an ABEND occurred. The ABEND is identified by a preceding IAMW73 error message, which contains the PSW and registers. Open processing for this file is terminated.

Return Codes: The OPEN fails with a return code of 8 and a reason code of 188(X'BC') in the ACBERFLG field of the ACB being opened.

Action: Contact Innovation Data Processing. Please have all of the available error messages available, along with any dumps to diagnose the problem.

IAMW25 DD='*ddname*' OPEN ERROR – INVALID COMPRESSION DICTIONARY

Reason: IAM was attempting to open a dataset with an internal customized hardware compression dictionary. IAM encountered an invalid or corrupted block that should have had a portion of the hardware compression dictionary.

Return Codes: The OPEN fails with a return code of 8 and a reason code of 188(X'BC') in the ACBERFLG of the ACB being opened.

Action: Attempt a dataset recovery, by either using IAMRECVR, or restoring the dataset from a backup. Before overwriting the corrupted dataset, back it up to tape and notify Innovation of the error.

IAMW26 DD='*ddname*' EXTEND CVAF READ (or WRITE) FAILED R15=xx STATUS=ddd

Reason: While IAM was expanding the size of a DFSMS Extended Format IAM dataset, the attempt to update the LSTAR in the VTOC failed. Refer to DFSMSdfp Diagnosis Reference for information on the return code and status value.

Return Codes: If an IAMDEBUG DD DUMMY dd card is in the affected job step, then the job step will abend with a U0184, otherwise IAM will continue processing the dataset.

Action: Contact Innovation with the error codes. If problem is repeatable, add in an IAMDEBUG DD DUMMY and a SYSABEND dd card to the job to get a dump at the time the error is detected.

IAMW30 DD='*ddname*' OPEN ERROR – FILE IN USE BY JOB jobname

Reason: While attempting to load an IAM file, it was determined that the IAM file is currently in use by some other job or an attempt was made to simultaneously load multiple IAM files to the same data set. IAM attempts to determine the name of the job using the data set, and will display the first job found that has the data set open in this message. If the owning job is not found, then *UNKNOWN is displayed.

Return Codes: A reason code of 192(x'C0') is stored in the ACB error flags field (ACBERFLG) and the open request is failed with a return code of 8.

Action: An IAM load must be executed without any other job accessing the file at the same time. Wait until the file is closed by the other users, then rerun the file load.

80.02 CONTINUED . . .

IAMW32 DD=DDNAME OPEN ERROR – JOURNALLING INITILIZATION FAILED

Reason: An error has occurred while attempting to open an IAM file that requested the use of the IAM Journalling exit. IAM was unable to complete initialization for journal processing. This message will be preceded by IAMJxx messages indicating the reason for the failure.

Return Codes: A reason code of 188 (x'BC') is stored in the ACB error flags field (ACBERFLG) and the open request is failed with a return code of 8.

Action: Review the accompanying IAMJxx messages for the corrective acdtion that is required.

IAMW33 DD=ddname EMPTY ALTERNATE INDEX, DSN=alternate index name

Reason: This error message occurs when an alternate index is being opened as an object of a PATH, or as part of an upgrade set, and OPEN determined that the specified alternate index had not been loaded. Valid loading of an alternate index is generally accomplished via an IDCAMS BLDINDEX function.

Return Codes: If the alternate index is the object of a PATH that is being opened, the open will fail with a return code of 8, and an error code of 196 (x'C4') in the ACBERFLG field. If the alternate index is being opened as part of an upgrade set, then the open receives a warning return code of 4, and an error code of 100 (x'64') in the ACBERFLG field.

Action: Load the identified alternate index data sets with an IDCAMS BLDINDEX, or other program with a comparable function. If the alternate index is no longer needed, make sure that it has been properly deleted.

IAMW34 DD=ddname IAM ADDRESS SPACE NOT ACTIVE
IAM RLS OPEN ABENDED

Reason: This error message may occur when attempting to open a file to the IAMRLS address space, and it was either not active, or it abended while processing this open request.

Return Codes: The open will fail with a return code of 8. If the IAMRLS address space is not active, the ACBERFLG field will be set to 182(x'B6'). If the IAMRLS address space abended, then the ACBERFLG field will be set to 167(x'A7').

Action: If IAMRLS is not active, then start it. If it abended, review the log and any dump(s) that may have been taken. Contact Innovation if assistance is required.

80.02 CONTINUED . . .

IAMW37 DD=ddname I/O ERROR ECB=xx CSW=xxxx SENSE=xxxx [op RBN=block]

Reason: This message is displayed when an I/O error has occurred processing an Enhanced Format IAM file, or during a load of an IAM file. This message may also be accompanied by an IEA000I/IOS000I error message. The message contains the following information:

ECB – displays high order byte of ECB as posted. Possible values include:

- 41–I/O terminated with error. CSW and / or Sense bytes are useful.
- 42–Block is not within DASD extents for data set.
- 47–DFSMS Extended Format block-id mismatch
- 4E–Attempted write of a block of all hex zeros.

CSW – contains the UNIT/CHANNEL status bytes from the CSW. Possible values include:

- 0C00–Normal Status
- 0E40–Unit Check and/or Wrong length record
- 0D40–End of file

SENSE – contains the first two sense bytes from the IAM internal IOB. Possible values include:

- 8000–Command reject, the device or control unit did not recognize the command
- 4000–Device requires operator Intervention
- 1000–Equipment check
- 0800–Data check
- 0040–Invalid Track Format
- 0020–End of cylinder
- 0008–Block not found

For file loads, the job step will be abended with a U0233 abend code.

For file access, additional information includes:

- op = RD error occurred on input operation or
- op = WR error occurred on output operation
- RBN= the relative block number being read or written

80.02 CONTINUED . . .

Action: If an IBM IEA000I/IOS000I message appears on the JCL LOG, a hardware error has occurred. The IBM message gives the sense information (ex: data check, equipment check). Examine the error information provided in the IBM message to determine the cause of the error. The format of this information is documented in the IBM data management SRL for the operating system in use. The file in question is unreadable in its present state and must be reestablished. This may be the result of a hardware error and if possible the new file should be allocated to a different physical location or volume.

For other I/O errors, those not accompanied by a hardware failure error message, determine the cause of the error. It is recommended that an IAMRECVR DIAGNOSE function be executed on the problem file, to see if there are any problems with the file integrity. Some common reasons for these types of I/O errors include:

- File has been improperly moved or restored to a device type different than it was originally loaded on. Frequently, this will fail with block not found I/O error, with a SENSE error code of x'0008'.
- Multivolume file has been improperly moved, or improperly cataloged. Frequently, this will result in an I/O error of block not within extent, ECB error code of x'42'.
- Storage overlay of IAM I/O control blocks. When this type of error occurs, most jobs and application programs do successfully process the IAM file, as does IAMRECVR. The failure is typically limited to one or a few jobs. A SYSUDUMP will be needed to determine the cause of this type of error.

The utility IAMRECVR may be used to recover a file that is no longer usable due to I/O or logical error conditions. This utility can be used to off load records from those portions of the file that have not been physically damaged. The user program should make a decision on the action to take when this occurs, for example to continue processing without this file or to terminate processing until this file is made available. The appropriate action will depend entirely on the application and the user program's evaluation of the diagnostic information that is returned. If further assistance is needed, contact Innovation Data Processing.

IAMW38 DD='*ddname*' DSPSERV CREATE FAILED, RC=xx REAS=nnnnnnnn

Reason: IAM attempted to create a data space for holding the index structure during the file load, however the request was rejected by MVS for the indicated return code and reason code. This is an informational message only, IAM will attempt to continue processing, and utilize a dynamically allocated temporary work file on DASD. For information on the return code and reason code, review in the IBM MVS/ESA Authorize Assembler Services Reference Manual, under the DSPSERV macro.

Return Codes: No error codes are set for this situation.

Action: Contact INNOVATION for assistance to resolve the problem if unable to do so after reviewing the return code and reason code provided. One of the common reasons for this error message is that the installation exit has either disallowed the use of data spaces, or limited the size. If the size is limited, reduce the IAM Global Option value for DATASPACE, or set it to 0 to prevent the use of data spaces.

IAMW39 DD='*ddname*' ###### RECORDS ACCEPTED PRIOR TO ABEND

Reason: IAM has determined that the task loading the specified file has abended, or hit some other type of error condition, after loading the indicated number of records. This information may be useful for determining how much to adjust the space parameters if some type of Sx37 abend has occurred. The number of records that were actually written to the file may be slightly less than indicated due to buffering. *Please note that the IAM file MUST BE RELOADED SUCCESSFULLY before attempting to otherwise access the file.*

Action: Correct the error condition as indicated by the abend, and rerun the job. Depending on the cause of the abend, the file may have to be deleted and redefined, for example if a larger space requirement is necessary.

80.02 CONTINUED . . .

IAMW40 CARD IMAGE –*cc.....cc*

Reason: The input control statement(s) read from the 'IAMOVRID' DD statement is displayed when the control statement(s) contains an error or if requested by the user via the 'LOG=YES' operand.

Action: If any error condition was raised, another message will indicate the reason for the error. Otherwise, no action is necessary.

IAMW41 CONTROL STATEMENT OPEN FAILED – DDNAME='*ddname*'

Reason: The DDNAME listed was required as control statement input to the override processor, IAMOVRID. An OPEN was attempted, but failed. Processing of the override service is terminated. Normal processing continues.

Action: Review the execution job log messages for more detail on cause of the failure. Correct the 'ddname' statement and, if necessary, rerun the job.

IAMW42 INVALID CONTINUATION CARD

Reason: User coded a delimiting comma following the last keyword on a control statement input to the override processor, IAMOVRID, and neglected to provide the next logical record. Processing of the override service is terminated. Normal processing continues.

Action: Correct the control statement. The job will continue to run, but may fail or perform unsatisfactorily if the Override data is critical.

IAMW43 I/O ERROR READING CONTROL STATEMENTS – DDNAME='*ddname*'

Reason: An I/O error occurred reading the data set referenced by 'ddname'. IAMOVRID is terminated, but normal processing continues.

Action: Examine any system message(s) to determine the cause of the error. The format of system messages is documented in the IBM MESSAGE SRL for the operating system in use. The job will continue to run, but may fail if the Override data is critical.

IAMW44 CONTROL STATEMENT BYPASSED – '*error description*'

Reason: An error was encountered by the override processor, IAMOVRID, during the processing of user supplied Override Control statements. The error description will be from the following list:

- DDNAME NOT SPECIFIED – The DDNAME operand was missing or misspelled. DDNAME is required to relate the override to a specific IAM file.
- MAXIMUM GLOBAL OVERRIDES EXCEEDED – The in storage table which holds the Overrides is full. A maximum of 200 control statements may be specified.
- INSUFFICIENT STORAGE – There was insufficient virtual storage for the IAM override processor to acquire for saving the internal format of the overrides.

Action: Correct the Override statements as follows:

- Add a corrected DDNAME operand to the control statement and, if necessary, rerun the job.
- Reduce the number of global overrides to 200 control statements or less. If more Override statements are required, contact INNOVATION technical support for a modification to IAMOVRID to expand the in-storage table.
- Increase the amount of region, particularly for above the line storage. The override processor needs 48K of storage for the override table. Additional storage will also be required for processing the data set being opened, so just an increase of 48K will most likely not be sufficient.

80.02 CONTINUED . . .

IAMW46 '*ddname*' OPEN FAILURE – IAMNINFO PROCESSING TERMINATED

Reason: The output report 'ddname' statement could not be opened by IAMNINFO. This DDNAME is usually 'IAMINFO', but may have been overridden by the user. Processing continues without interruption with the IAMINFO report bypassed.

Action: Correct the allocation of DDNAME 'ddname' so the next execution of the job will produce the IAMINFO report.

IAMW47 I/O ERROR MONITORING DSN – '*dsname*' – IAM MONITOR TERMINATED

Reason: An IAM monitor facility processor encountered an I/O error while writing monitor/trace data. Monitoring has been discontinued. Normal IAM processing continues.

Action: If a Monitor report is needed, correct the cause of the I/O error and rerun the job.

IAMW48 IAMNINFO PARAMETER LIST ABSENT OR IN ERROR – PROCESSING TERMINATED

Reason: The parameter list required by the dynamic file status display processor, IAMNINFO, was missing, was overlaid or is in error. This maybe an internal error. Processing continues without the IAMINFO reported printed.

Action: If you are unable to determine the reason for the message, call INNOVATION for further assistance.

IAMW50 IAM VTOC ACCESS FAILED COMP=xxxx CODE=xxxx CLUSTER=clustername

Reason: During an IDCAMS DEFINE or RECATALOG of an IAM file, an attempt to access or update a VTOC failed. The completion code and return codes from CVAF are displayed in the message.

Action: Refer to the IBM manual 'Common VTOC Access Facility Diagnosis Reference' for the meaning of the codes. Correct the problem and re-submit. If unable to correct the problem, contact INNOVATION for assistance.

IAMW51 IAMOVRID CONTROL STATEMENT ERROR, CLUSTER=dsname

Reason: During an IDCAMS DEFINE of an IAM file, the IAM Override processor detected an error or invalid IAM override card.

Return Codes: The DEFINE is failed with a return code of 140 and a reason code of 36. These codes will appear on the IDC3009I message produced by IDCAMS. The DEFINE is failed on an override error to prevent IAM from defining a file with incorrect attributes.

Action: Correct the error on the IAM Overrides, and rerun the dataset. Review Section 30 for proper IAM Override parameters.

IAMW52 IAM SHOWCAT INTERCEPT FAILED

Reason: The IAM SHOWCAT intercept function within VIF failed.

Return Codes: The calling program will be abended with a U0283 abend code. This error would indicate that something has destroyed the IAM VSAM interface table in virtual storage.

Action: Obtain a SYSABEND dump and call INNOVATION for assistance.

80.02 CONTINUED . . .

IAMW53 RECAT FAILED, DATASET NOT CATALOGED CLUSTER=clustername
DATASET NOT ON VOLUME: volserCLUSTER=clustername
OBTAIN FAILED, VOLUME: volserCLUSTER=clustername
DATASET FAILED IAM VALIDATIONCLUSTER=clustername

Reason: An error as indicated by the reason in the message occurred during an IDCAMS DEFINE RECATALOG of an IAM dataset.

Return Codes: The RECATALOG request will be failed with return codes from the failing service.

Action: Make the corrections to the RECATALOG to correct the identified error, and rerun. If unable to determine why the request failed, contact INNOVATION for assistance.

IAMW54 PROBABLE IAM FILE HAS NOT BEEN DEFINED, DSN=dsname

Reason: The processing program issued a SHOWCAT catalog request for a non-VSAM file that is cataloged, but does not have the IAM information (as established by DEFINE or file load) was not returned. This message is for diagnostic purposes, and will only be issued when an IAMDEBUG DD DUMMY DD card is in the job step of the program issuing the SHOWCAT macro.

Return Code: The SHOWCAT is given a return code of 32 (x'20'), indicating that the file can not be accessed through IAM or VSAM.

Action: This is an unexpected error situation, and should be reported to Innovation Data Processing. Please have a LISTCAT ALL output from IDCAMS available when calling.

IAMW56 IAM DEFINE OF NON-SUPPORTED FILE TYPE CLUSTER=dsname

Reason: An IDCAMS DEFINE was issued with an indication that the file should be an IAM file, but the file type cannot be converted to IAM. Examples are VSAM RRDS (Relative Record data sets) and VSAM Linear data sets.

Return Codes: The DEFINE is failed with a return code of 22, and a reason code of 8. These codes will appear in an IDC3009I message.

Action: Correct the IDCAMS DEFINE to either change the file type to one supported by IAM, or remove the indication that the file is to be an IAM file.

80.02 CONTINUED . . .

IAMW57 IAM ALLOCATION FAILED COMP=xxxx CODE=xxxx CLUSTER=clustername

Reason: An error occurred during an IDCAMS DEFINE of an IAM file. The codes displayed correspond to the return code and reason codes of the IDC3009I message from IDCAMS, and there will also be a IDC3009I message on SYSPRINT with the same codes. There may also be additional IDC or IAMW messages.

Return Codes: The DEFINE request is failed with the return code and reason code given in this message. Some of the more common return codes include:

- 8,38—Data set already cataloged
- 16,0—SMS failed allocation request, refer to IGD messages
- 22,8—IAM does not support type of VSAM file requested.
- 42,0—MVS DADSM failed allocation request
- 56,6—User not RACF authorized to define the file
- 58,0—Obtain of VTOC entry failed
- 58,4—Specified DASD volume(s) not online
- 68,20—No space on selected volume
- 140,36—Invalid IAM Override card
- 176,0—No space in VTOC
- 184,4—Data set is allocated to another job or user
- 192,0—Exceeded maximum allowable IAM record size

Action: Refer to section 80.20 Catalog Return Codes or IDCAMS error message IDC3009I for meaning of the codes. Correct the problem, and resubmit. It may be necessary to issue an IDCAMS DELETE command before attempting to resubmit the DEFINE.

IAMW58 LISTC INTERCEPT FAILED COMP=xxxx CODE=xxxx CLUSTER=UNKNOWN

Reason: A catalog Locate or LISTC request intercepted by IAM failed and received the specified completion and return codes.

Return Codes: The request is failed with the indicated return code and reason code.

Action: Refer to VSAM errors message IDC3009I for meaning of the codes. Correct the problem and resubmit.

IAMW59 LOCATE FOR AN IAM FILE FAILED CODE=(cc)xxx [DSN=......]

Reason: Locate for an IAM file failed for one of the following reasons:

1. CODE=S1xxx The IAM SHOWCAT intercept issued a locate which failed with return code xxxx on the specified data set. This form of the message will only appear when there is an IAMDEBUG DD DUMMY coded in the failing job step.
2. CODE=S2xxx The IAM SHOWCAT intercept issued a locate which failed with return code xxxx. The locate was issued by CI number, so the data set name is unknown.

Return Code: The SHOWCAT is given a return code of 32 (x'20'), indicating that the file can not be accessed through IAM or VSAM.

Action: Make sure that the data set is still properly cataloged. If not, an IDCAMS DEFINE RECATALOG must be done. If further assistance is required, contact INNOVATION.

80.02 CONTINUED . . .

IAMW60 IAM DYNALLOC FAILED '*description*'

Reason: During the processing of an IDCAMS DEFINE for an IAM file, IAM's attempted use of Dynamic Allocation failed for the specified reason. This message presents a brief English description of the error code returned by Dynamic Allocation, which is supplied in the IAMW61 error message. Both messages are printed on the system log. There will also be an IDC3009I message on SYSPRINT, with an appropriate error code.

Return Code: The DEFINE request is failed, with a return code and reason code that matches the problem description.

Action: Correct the error situation, as described with the matching text below, and rerun the DEFINE.

Text: **DATA SET NAME IN USE BY ANOTHER JOB/USER**

Reason: The dataset name has been enqueued on by another job/user.

Action: Through whatever software facilities available, determine which job and/or users are enqueued on the data set, and rerun the DEFINE upon the completion of the other job/user.

Text: **VOLUME NOT MOUNTED ON SPECIFIED UNIT**

Reason: The specified volume was either not mounted, or was mounted but not on the unit specified by the UNIT= keyword on the IAM override control statement for this file. For non-specific volume requests, (i.e., with VOL(ANYVOL) coded), there were no volumes mounted as storage for the unit name specified on the IAM Override Control statement, or SYSDA.

Action: Mount the required volume, or change the volume and/or unit specification.

Text: **SPECIFIED UNIT NAME IS UNDEFINED**

Reason: The unit name specified on the IAM Override Control statement for this file does not exist on the system that the define was attempted.

Action: Correct the unit name specification, or run on the proper operating system.

Text: **REQUIRED CATALOG NOT MOUNTED**

Reason: The catalog required for the definition of the IAM data set is on a volume that is not currently mounted.

Action: Insure that the volume containing the user catalog is mounted, and rerun the DEFINE.

Text: **DATA SET ALREADY EXISTS**

Reason: The data set being Defined already exists in the catalog, and may or may not be on the volume it is cataloged to.

Action: Make sure that the cluster name is correct and if not correct it. If it is correct, delete the data set from the catalog (and volume if applicable) and rerun the DEFINE.

Text: **DUPLICATE DATA SET NAME ON VOLUME**

Reason: The data set already exists on the specified volume, and is not cataloged.

Action: Delete the data set from the volume, and rerun the DEFINE.

Text: **NO SPACE IN VTOC**

Reason: There was no space in the VTOC (Volume Table of Contents) or the VTOC index for the new data set on the specified or selected volume.

Action: Either correct the error by increasing the size of the VTOC or VTOC Index on the volume (this can be done by use of COMPAKTOR), delete unwanted data sets from the volume, or select a different volume.

80.02 CONTINUED . . .

Text: **VTOC I/O ERROR OR CVAF ERROR**

Reason: An I/O error occurred on the VTOC during file allocation.

Action: Review SYSLOG for other messages indicating a more precise cause of error. Correct the problem and rerun DEFINE.

Text: **REQUESTED SPACE NOT AVAILABLE ON VOLUME**

Reason: The volume specified or selected did not have sufficient space to satisfy the request.

Action: Ensure that the space requested is actually needed, and adjust if possible. (NOTE: IAM files generally require less space than VSAM files.) Otherwise, select a different volume, remove unneeded data sets from the volume, or run COMPAKTOR to consolidate free space.

Text: **USER NOT AUTHORIZED TO ALLOCATE DATA SET**

Reason: The job lacks RACF authorization to DEFINE the data set.

Action: Contact the Security Administrator for assistance.

Text: **INSTALLATION EXIT REJECTED ALLOCATION REQUEST**

Reason: A dynamic allocation exit routine in the system did not allow the allocation request to be processed.

Action: Correct the DEFINE to the installation requirements.

Text: **REQUIRED CATALOG NOT AVAILABLE**

Reason: The user catalog required may have been Deleted or disconnected from the system master catalog, or may have been damaged and is being recovered.

Action: Correct the error with the user catalog, and rerun the DEFINE command.

Text: **DUPLICATE DATA SET NAME IN CATALOG**

Reason: The data set name already exists in the catalog, and may or may not exist on disk.

Action: Make sure the cluster name is correctly specified. If it is delete the current entry from catalog (and disk if applicable).

Text: **NO SPACE IN CATALOG**

Reason: Insufficient space in the catalog to contain the record for the new data set.

Action: Enlarge the catalog, and rerun the DEFINE.

Text: **SMS FAILED REQUEST. REFER TO PRIOR MESSAGE(S)**

Reason: The allocation request was failed by SMS. There should be preceding messages from SMS indicating the reason for the error.

Action: Correct the problem indicated by the SMS error messages, and try request again.

80.02 CONTINUED . . .

IAMW61 IAM DYNALLOC FAILED COMP=nnnn CODE=nnnn CLUSTER=clustername

Reason: The dynamic allocation requested by IAM to perform the DEFINE operation failed with the printed error codes. This message may be accompanied by an IAMW60 message.

Return Code: The DEFINE request is failed, with a return code and reason code that matches the problem description.

Action: Refer to message IAMW60, if printed, and/or the IDC3009I error message on SYSPRINT. The error codes from Dynamic Allocation are documented in the MVS/XA and MVS/ESA System Macro and Facilities manual, the MVS Job Management SPL, and under the ISPF tutorial. Correct the error condition as indicated by the error codes, and rerun the DEFINE.

IAMW62 IAM OPEN FAILED FOR DDNAME=ddname CLUSTER=clustername

Reason: During DEFINE processing of an IAM file, IAM attempted to OPEN the defined file, however the OPEN failed. Additional IBM messages may appear on the system log.

Return Codes: The DEFINE request is failed with a return code of 62, reason code of 0.

Action: Determine the cause of the OPEN failure, correct the error, and rerun the job. For a new DEFINE (as opposed to RECATALOG), DELETE and redefine the data set.

Note: At this point, the data set has been allocated and cataloged, but is not yet usable by IAM.

IAMW63 IAM I/O ERROR: '*synad message*'

Reason: During the processing of a DEFINE command for an IAM file, an I/O error occurred when reading or writing the IAM control information.

Return Codes: The DEFINE request is failed with a return code of 62, reason code of 0. If an IAMDEBUG DD DUMMY is specified, then the program will abend with a U0310.

Action: Using the standard SYNAD message and other messages that may appear on SYSLOG, determine the cause of I/O error and correct it. If this was not a RECATALOG operation, DELETE and DEFINE the IAM data set again.

Note: At this point, the data set has been allocated and cataloged, but is not yet usable by IAM. For RECATALOG operations, the file is either not a previously DEFINED or loaded IAM file, or there is an error with the data set requiring recovery. The recovery can be done by restoring the data set from a good copy or possibly by using program 'IAMRECVR'. Use of the recovery program may result in data loss.

IAMW64 UNEXPECTED END OF FILE READING AN IAM FILE FOR RECATALOG REQUEST – NOT VALID IAM DATA SET

Reason: During the processing of a DEFINE RECATALOG command for an IAM file, an end of file occurred while attempting to read the file characteristics.

Return Codes: The DEFINE request is failed with a return code of 86, reason code of 4.

Action: The data set is empty. The recatalog request was not performed. Either the file was never an IAM file, in which case no corrective action is required, or the data set has been clobbered. To recover the data set, it can be restored from a good backup, or a recovery attempted with program 'IAMRECVR'. Recover the file, then retry the recatalog processing.

80.02 CONTINUED . . .

IAMW65 IAM SCRATCH FAILED COMP=nnnn CODE=nnnn VOLSER=vvvvvv
IAM UNCATLG FAILED COMP=nnnn CODE=nnnn CLUSTER=clustername

Reason: After an error attempting to catalog or initialize an IAM file being defined, an attempt to delete or uncatalog the data set failed. The codes are returned from SCRATCH or UNCATLG request, which are documented in the SYSTEM DATA ADMINISTRATION manual.

Return Codes: The return code for the DEFINE is based on the original condition that caused the error.

Action: The data set is still on the specified volume. Refer to an immediately preceding IAMWnn message for the data set name. The data set must be manually scratched from the indicated volume and uncataloged prior to attempting to redefine the data set.

IAMW66 IAM REALLOC FAILED CODE=nnnn INFO=nnnn CLUSTER=clustername

Reason: After successfully defining an IAM file, IAM had determined that the job step had DD cards which were allocated to the file, but were allocated to the wrong volume. The attempt to reallocate the file with dynamic allocation failed, with the indicated error codes.

Return Codes: The DEFINE completes with a return code of 0.

Action: The IAM file has been successfully defined, but attempts to REPRO into the IAM file within the same step may fail. A subsequent REPRO into the IAM file can be done.

IAMW67 IAM SMS ALLOC FAILED RC=X'*xx*' REAS=X'xxxxxxxx' CLUSTER=clustername

Reason: The define of an IAM file failed using DADSM allocation with the specified return code and reason code. Refer to the DADSM Create (ALLOCATE) Function Return Codes section of the *IBM MVS/ESA DADSM/ CVAF Diagnostic Aids* for a description of the error codes.

Return Codes: The DEFINE request is failed with a return code of 62, reason code of 0.

Action: Take the appropriate corrective action based on the error codes indicated, and retry the define request.

IAMW68 IAM UNIT NAME SEARCH FAILED, RC=xx CLUSTER=clustername

Reason: During the define of a multivolume nonspecific allocation, an IAM call to the MVS Unit Name look up service failed as indicated in the message. The return code, if provided, is documented in the *IBM MVS System Modifications Manual.*

Return Codes: The DEFINE request fails with a return code of 72, reason 4.

Action: If the condition indicated by the return code cannot be corrected, contact INNOVATION for support. As a circumvention, try a different UNIT override, or switch to specific volume allocation.

IAMW69 IAM xx ELIGIBLE VOLUMES, nn VOLUMES REQUIRED CLUSTER=clustername

Reason: During the define of a multivolume nonspecific allocation, IAM found the indicated number of storage volumes in the specified UNIT name pool, however, more volumes then available were needed to satisfy the allocation request.

Return Codes: The DEFINE request fails with a return code of 72, reason 4.

Action: Change the UNIT override to indicate a UNIT name that has sufficient storage volumes, or reduce the number of volumes requested.

80.02 CONTINUED . . .

IAMW70 DD='*ddname*' PUT ERROR – WORK FILE DATASPACE OVERFLOW

Reason: The size of the data space used to hold the index structure during a file load was insufficient for the file indicated by the ddname.

Return Codes: A reason code of 244(x'F4') is stored in the RPL error flags field (RPLERRCD) and the request fails with a return code of 8. The logical error exit, LERAD, if specified will be invoked. If the error is detected during close, the program is abended with a U0246 abend.

Action: Increase the size of the data space by using the IAM CREATE override, specifying the DATASPACE keyword. As an alternative, specify a DATASPACE=0, which will force the use of a temporary work file.

IAMW71 TRACE DEACTIVATED – text indicating reason

Reason: The IAM trace facility for Enhanced Format files has detected an error during activation. The possible reasons include

- The DDNAME IAMATR31 is not available
- There is insufficient storage available to obtain the trace work area.

Action: If the IAMATR31 DD was not specified, add it to the job. If storage was not available for the trace work area, specify a larger REGION size, or a REORG of the IAM file may be required.

IAMW72 IAMASY ESPIE RECOVERY ENTERED FOR ABEND S0Cx

Reason: A program check occurred under the IAM IRB while performing asynchronous processing for an I/O request with RPL OPTCD=ASY. Included in the displayed information are the PSW and registers at time of the error.

Return Codes: The specific request causing the error, if it is identified, will receive a return code of 8 with an RPL error code of 240(X'F0') stored in the RPLERRCD. IAM will internally issue an ENDREQ for that RPL to clean up any resources it may have held.

Action: IAM will attempt to continue processing for the affected file. Contact Innovation Technical Support with the full text of the message for assistance, and save any dumps that may have occurred to aid in diagnosis.

Note: This message will not occur in IAM Version 8.0. It is presented here for documentation compatibility with prior versions.

IAMW73 IAMAVSOC ESTAE ENTERED FOR ABEND Sxxx

Reason: A abend occurred while opening an IAM file. IAM will attempt to free the resources acquired during the open process. Depending on the circumstance, the PSW and registers at time of the error may also be displayed.

Return Codes: The job will abend, except if the open occurred under CICS or for a TSO user. CICS and TSO users will abend with a U0184 only if there is a //IAMDEBUG DD DUMMY DD card allocated. Otherwise, CICS and TSO will have the open fail with a return code of 8, and the ACB that has caused the error will have a reason code of 188(X’BC’) set in the ACBERFLG field. If IAM was using a Data Space for this file, it will be included in the dump.

Action: Contact Innovation Technical Support with the full text of the message for assistance, and save any dumps that may have occurred to aid in diagnosis. For CICS, IAM will terminate open processing for this ACB, and will attempt to free all resources used by the failing OPEN request.

80.02 CONTINUED . . .

IAMW74 UCBLOOK MACRO FAILED R15=xxxx R0=xxxx CLUSTER=clustername

Reason: During IAM file definition, an attempt to find the UCB on which the data set resides, or will reside, using the IBM UCBLOOK service has failed. IAM will continue, if possible, with the define request by utilizing a different UCB lookup technique.

Action: Contact Innovation Technical support with the full text of the message for assistance.

IAMW78 DD='*ddname*' OPEN ERROR – AN ABEND CONDITION OCCURRED

Reason: While IAM was opening a file to be loaded, some type of system Abend occurred. There should be additional messages indicating the exact abend condition. Open processing for this file is terminated.

Return Codes: The OPEN fails with a return code of 8 and a reason code of 192(X'C0') in the ACBERFLG field of the ACB being opened.

Action: Correct the abend condition, and then reload the file. The most typical abend condition is an X37 abend, in which case delete and redefine the data set with more DASD space.

IAMW79 DD='*ddname*' OPEN ERROR – text indicating reason

Reason: IAM was unable to open the specified file due to damaged control or index information on the file.

Return Codes: A reason code of 188(X'BC') is stored in the ACB error flags field (ACBERFLG) and the OPEN is failed with a return code of 8. If an IAMDEBUG DD DUMMY is in the job step, then the job will be abended with a U0184.

Action: ***DO NOT ATTEMPT TO REORGANIZE THE DATA SET WITH IDCAMS OR OTHER SOFTWARE!!!*** **Use the IAMRECVR to recreate the file.** For diagnosis, please do the following:

1. Rerun job with an //IAMDEBUG DD DUMMY card added and a //SYSUDUMP or //SYSABEND. If you have the ABENDAID, please insure that the standard IBM dump will be taken.
2. Save the damaged file, or back it up using FDR/ABR, or DFDSS.
3. Contact Innovation Data Processing for assistance.

IAMW80 MODULE '*modname*' INSTALLED AT '*address*' – VER nn

Reason: The message is issued in response to a VIF status request when the module 'modname' is in place and is ready to provide IAM services to programs using ACB's to access VSAM files. The 'address' given is the virtual storage location of the named IAM VIF module. The version level number is also given for the module.

Action: None, information message only.

80.02 CONTINUED . . .

IAMW81 THE IAM SYSTEM MODULES ARE '*status*'

Reason: This message is provided by IAMSTART when the IAM VIF modules are installed, or in response to VIF action commands when the IAM system level VSAM interface is already in place. The 'status' of the modules may be:

- ACTIVE
- REACTIVATED
- ALREADY INSTALLED

This message includes the version and release level of the VIF modules installed.

Action: None, information message only.

IAMW82 THE IAM SYSTEM MODULES ARE '*status*'

Reason: This message is provided by IAMSTART in response to VIF action commands when the IAM system level VSAM interface is already in place. The status of the modules maybe:

- NOT ACTIVE (the IAM system level VSAM interface is not in place),
- INACTIVE (the IAM system level VSAM interface is in place but is not active).

Action: None, information message only.

IAMW84 IAMSTART ESTAE RECOVERY ENTERED FOR ABEND Sxxx Uxxxx AT OFFSET xxxx

Reason: This message is provided by IAMSTART when VIF ABENDS. An attempt is made to provide diagnostic information.

Action: If the problem persists call INNOVATION for assistance.

IAMW85 'ADDRESS' 'HEX core print'—- 'EBCDIC core print'

Reason: This message is provided by IAMSTART when the VIF modules are first installed and is also the response to a VIF STATUS request when VIF is in place and is ready to provide IAM services to programs using ACB's. The 'ADDRESS' given is the virtual storage location of the VIF Vector Table Entry. The remainder of the line is the entry in hex and display format.

Action: None, information message only.

80.02 CONTINUED . . .

IAMW86 IDPSTART FAILURE – REASON='*number*' '*reason*'

Reason: The activation of the IAM VSAM Interface failed for one of the following reasons:

- 01—OPERATING SYSTEM NOT MVS OR SP 1.2 OR HIGHER
- 02—INVALID INPUT PARAMETERS
- 03—UNABLE TO OBTAIN AUTHORIZATION
- 04—'vector table name' VECTOR TABLE IS INVALID
- 05—SYSLIB DD MISSING OR OPEN ERROR
- 06—'modname' NOT FOUND IN SYSLIB
- 07—'modname' MODULE LOAD ERROR
- 08—ERROR MODIFYING THE SVC TABLE
- 09—'modname' NOT FOUND
- 10—'modname' NOT FOUND IN THE LINKLIST
- 11—'modname' UNABLE TO DE-INSTALL
- 12—'modname' CDE OR LPDE ABOVE 16M ERROR
- 13—'modname' MODULE ABOVE 16M ERROR
- 14—GETMAIN ERROR
- 15—'modname' HAS SMP INSTALLED IDP MODULE
- 16—UNABLE TO OBTAIN DISPATCHER LOCK
- 17—FREEMAIN ERROR
- 18—UNABLE TO OBTAIN LOCAL LOCK
- 19—'modname' CDE NOT FOUND
- 20—ENQUEUE/DEQUEUE ERROR ON IAM RESOURCE
- 21—ENQUEUE/DEQUEUE ERROR ON SYSZSVC
- 22—INVALID VECTOR TABLE STATUS
- 23—INVALID VECTOR TABLE CDE ADDRESS
- 24—RC=xxxx FROM SVCUPDTE
- 27—'modname' and IAMVECTB ARE OUT OF SYNCH
- 28—VECTOR TABLE CANNOT BE STOPPED
- 31—HIGHER VERSION OF VIF ALREADY STARTED

Action: If the problem persists call INNOVATION for assistance.

80.02 CONTINUED . . .

IAMW89 IAM – TRIAL VERSION FROM INNOVATION DATA PROCESSING EXPIRES IN '*nnn*' DAYS (PLEASE CONTACT INNOVATION)

Reason: This is a trial version of the IAM system. The number of days the trial will remain active is displayed.

Note: This message will never appear if you are a licensed user of IAM. If you are a licensed IAM users, then it is possible that you have in your job a bad STEPLIB pointing to the old trial library. The library that contains the production copy will appear in the heading with a P following the version identification. For example, IAM V8.0/04P.

Action: When there are 10 or fewer days before the trial is due to expire this message will become nondeletable. To prevent the trial from expiring call INNOVATION for an extension PARM= value and use the JCL shown below to extend your trial's expiration date. The JCL to extend the trial is as follows:

```
//EXTEND      EXEC    PGM=IAMEXTND,PARM=xxxx
//STEPLIB      DD     DISP=SHR,DSN=your.user.lib
//SYSLIB       DD     DISP=SHR,DSN=your.user.lib
//@BINDNOT     DD     DUMMY
//SYSDIAG      DD     SYSOUT=A
```

IAMW90 IOSCAPU MACRO FAILED COMP=xxxx CODE=yyyy

Reason: IAM attempted to capture a UCB to initialize the IAM file being defined. The capture of the UCB failed, as indicated in the error message.

Action: IAM terminates processing for the file being defined. The file will need to be deleted and redefined before it can be processed. The error codes are available in the MVS Authorized Assembler Services Reference manual. If assistance is needed with diagnosing the problem, contact Innovation Data Processing. Attempting to define the file an a different volume(s) may circumvent the problem. A dump can be obtained by including an IAMDEBUG DD DUMMY statement in the JCL, which will result in a U0310 abend.

IAMW99 INTERNAL LOGIC ERROR – JOB TERMINATED

Reason: IAMOVRID has encountered an illogical condition.

Action: Obtain a SYSUDUMP or SYSABEND dump, and call INNOVATION for assistance.

80.03 IAM SYSTEM SYSPRINT MESSAGES

The following general messages are issued by various IAM utility programs. These messages are typically written to SYSPRINT by the originating utility program. These are messages from the IAMRECVR, IAMSIMVS, IAMSMFVS, IAMSMF, and IAMZAPOP utility programs provided with the IAM product.

IAM099 NEAR RELATIVE DATA LOCATION nnnn '*error description*'

Reason: The common parsing routine encountered an error in parsing the user specified control statements. The approximate location of the error was position nnnn, counting the first position as 000.

Action: Correct the error and resubmit job.

IAM100 IAM FILE ANALYSIS – DSN= data set name

Reason: This report is produced from an LISTCAT request on the IAMPRINT DD output, and also on SYSPRINT when IAMRECVR is run against an IAM file. The report contains descriptive information and statistics about the IAM file. For a complete description of the output report following this message, please refer to the LISTCAT Report description section in the IAM manual.

Action: None, this is an informational message only.

IAM213 'file type' RECOVERY FILE CREATED DDNAME='*ddname*'

Reason: IAMRECVR has completed the output of a file to the DDNAME 'ddname'. The 'file type' is as follows:

IAM—An IAM file created by the RECOVER operation.

DUPLICATE—A key/log file containing duplicate records found in the IAM file that is being recovered by IAMRECVR.

Action: Refer to the documentation of IAMRECVR for the uses of the different output files it creates.

IAM260 LISTC PROCESSING FAILED FOR DSN='*dsname*'

Reason: An error occurred during IAM processing of an IDCAMS LISTC command for a possible IAM file. This message will be preceded by one or more error messages describing the error in more detail.

Action: Review other messages for source of error, and correct as necessary. IDCAMS LISTC processing will continue normally, however no information will appear in the IAMPRINT file for the named file.

IAM262 DATA SET IS NOT AN IAM FILE

Reason: During an IDCAMS LISTC IAM attempted to process a data set as an IAM file, however it was determined that the data set was not an IAM file. The data set name is displayed in the IAM260 message.

Action: None, information message only.

IAM266 LOAD OF MODULE '*modulename*' FAILED

Reason: During an IDCAMS LISTC command, IAM encountered an error loading the named support module. There should also be an accompanying message in SYSLOG indicating the cause of the problem (i.e. S106, S306, or S806 error). The data set name is presented in the accompanying IAM260 message.

Action: Find the IBM error message, and take corrective action. Possibilities are insufficient virtual storage to load the module, or IAM is not in the system LINKLIST and no STEPLIB has been provided.

80.03 CONTINUED . . .

IAM269 IAM CPL PROCESSING FAILED

Reason: IAM was attempting to process a CATALOG parameter list to determine data set name and volume information, however the CPL did not contain the expected information. There will be no information listed about the IAM file.

Action: Contact INNOVATION for assistance.

IAM303 CARD IMAGE – * control statement image *

Reason: A display of the SYSIN data set input control statements.

Action: None. Information message only.

IAM316 RECOVERY CAN BE FORCED BY SPECIFYING VALUES FOR THE FOLLOWING –

Reason: An attempt was made to recover an IAM file that has damaged control records. The unreadable control records requires the user to specify key data normally extracted from the file.

Action: The next line(s) displayed will detail the fields required. Use this information to continue the recovery.

IAM318 * WARNING* DATASET–'*dsn*' DEFINED FIXED CONTAINS VARIABLE LEN RECORDS

Reason: The DEFINE for this cluster shows the average and maximum record lengths to be equal. IAMSIMVS detected one or more records that were not equal to the average record length Defined for this cluster.

Action: IAMSIMVS continues processing as if the file contained fixed length records. Blocking, overflow and other values established for a file however vary depending upon whether record lengths are fixed or variable. To obtain a more accurate estimate of IAM's space savings for this file, include the 'VARIABLE' keyword with the 'SELECT' option.

Note: To properly identify this file as containing variable length records the DEFINE for this file should be changed so the average RECORDSIZE value is less than the maximum. During processing, if the Defined average RECORDSIZE value is not changed, IAM will return a record length error for this file.

IAM319 PREMATURE END OF FILE – AFTER BLOCK nnnnnnnn

Reason: IAMRECVR detected a premature end of file at block nnnnnnn. Data blocks may have been lost.

Action: Review the contents of the recovered file. A section of the file being recovered may not have been readable.

IAM320 MAXIMUM BLOCKS LOST DUE TO END OF FILE – nnnnnnnn

Reason: An end of file error erases the remainder of the track. The reported number of blocks could have existed on the track but were not yet read when the end of file was encountered.

Action: Review the contents of the recovered file. A section of the file being recovered may not have been readable.

80.03 CONTINUED . . .

IAM321 INVALID VARIABLE LENGTH FIELD – BLOCK nnnnnnnn

Reason: IAMRECVR detected an invalid RDW length field for a given record (i.e.: RDW exceeds maximum LRECL). The block number is printed and the remainder of block is bypassed.

Action: Review the contents of the recovered file. A section of the file being recovered may not have been readable.

IAM322 DROPPED DUE TO I/O ERROR – BLOCK nnnnnnnn

Reason: IAMRECVR has dropped block nnnnnnnn. The first 24 bytes of the block at the location of the errors is printed in hexadecimal. Processing continues.

Action: Review the contents of the recovered file. A section of the file being recovered may not have been readable.

IAM323 SEQUENCE CHECK – BLOCK nnnnnnnn

Reason: An out of sequence record was encountered. The IAM block number is printed along with 24 bytes of the key in hexadecimal. Processing continues.

Action: Review the contents of the recovered file. A section of the file being recovered may not have been readable.

IAM324 UNABLE TO CALCULATE LOST BLOCKS – ASSUMING nn MISSING

Reason: An end of file erases the remainder of the track. Any blocks that were on the track at the time the EOF record was created have been destroyed.

Action: Review the contents of the recovered file. A section of the file being recovered may not have been readable.

IAM325 FILE NOT COMPRESSABLE – DATA LEN PAST KEY < 11 BYTES – DSN=

Reason: IAMSIMVS determined that the records contained within the file were not eligible for IAM record compression. IAM only compresses the data in a record located past the end of the key and only if the length of that data is more than 10 bytes. (i.e. MAXLRECL-(RKP+KEYLEN) > 10).

Action: No action is required. IAMSIMVS will continue the simulation for the file as if it were not converted to an IAM file in compressed format.

IAM325 ERROR DECOMPRESSING RECORD – BLOCK nnnn

Reason: IAMRECVR was unable to decompress a compressed record from an IAM data set.

Action: Most probable cause is that a corrupted buffer was written out to the data set. If possible, recover the file through other means. If that is not possible then continue IAMRECVR will recover all the records that it is able to read, but will drop the rest of the records. Backup the data set as described in the Users Guide section of the IAM Manual, and contact Innovation for support.

80.03 CONTINUED . . .

IAM326 IAMCRTSM ERROR – 'error description' – DSN=dsn

Reason: IAMCRTSM was invoked to simulate the conversion of a VSAM file and encountered a logical error. The error description will be in the format: INVALID BLOCKSIZE/BLOCKING FACTOR CODE=nnn. The CODE=nnn translates directly into an IAM create abend code (see IAM user abend, error, and completion codes later in this section).

Action: If the message was issued for invalid blocking, ensure that the blocksize specified is both larger than the record length and compatible with the device. If the CODE= message is issued, refer to the actions for the associated IAM create abend code.

IAM340 'recovery operation' – DSN= '*dsname*'

Reason: Describes the user specified IAM file 'recovery operation' and the name of the IAM file being processed.

Action: None, information message only.

IAM341 CONTROL BREAKS IN OVERFLOW BLOCKS – = nnnnnnnn

Reason: Issued by the DIAGNOSE and RECOVER operations of the IAM file recovery program IAMRECVR, indicating that records within the overflow blocks of the file being processed are out of sequence.

Action: When issued during RECOVER, it indicates that the RECOVER output is out of sequence and must be sorted, either by IAMRECVR or externally prior to an IAM create. When issued during DIAGNOSE, no additional processing is necessary.

IAM342 IAM FILE CONTAINS NO DETECTABLE ERRORS

Reason: Issued by the DIAGNOSE and RECOVER operations of the IAM file recovery program IAMRECVR, indicating that the IAM file is not damaged and is acceptable for IAM processing.

Action: None, information message only.

IAM343 SEQUENCE ERRORS IN DATA BLOCKS – = nnnnnnnn

Reason: Issued by the DIAGNOSE and RECOVER operations of the IAM file recovery program IAMRECVR, indicating physical damage to the IAM file.

Action: The file must be created again from a suitable backup or recovered and reconstructed. Prior to create, sort the data set output from the RECOVER operation.

IAM344 NUMBER OF DROPPED BLOCKS – = nnnnnnnn

Reason: Issued by the DIAGNOSE and RECOVER operations of the IAM file recovery program IAMRECVR, indicating physical damage to the IAM file. IAMRECVR encountered I/O errors during the processing of the IAM file and nnnnnnnn blocks were dropped from the file.

Action: The file must be created again from a suitable backup or recovered and reconstructed. Prior to create, sort the data set output from the RECOVER operation.

80.03 CONTINUED . . .

IAM345 NUMBER OF DUPLICATE RECORDS – = nnnnnnnn

Reason: Issued by the DIAGNOSE and RECOVER operations of the IAM file recovery program IAMRECVR, indicating that duplicate keys were found in the IAM file. During a RECOVER operation, duplicate records can be ignored, printed, logged for subsequent application to the file, or applied directly if an IAM file is being created as the output of the recovery program.

Action: See the documentation for the RECOVER program, DUPLICATES operand. The file must be created again from a suitable backup or recovered and reconstructed. Prior to create, sort the data set output from the RECOVER operation.

IAM346 SPANNED RECORDS EXTRACTED – = nnnnnnnn

Reason: Issued by the IAMRECVR RECOVER command when spanned records have been recovered and placed in the dataset identified by the SPANOUT DD statement.

Action: To recover the spanned records, run the IAMRECVR APPLY SPANNED command, which will read the SPANOUT dataset, and insert the records back into the IAM data set. See Section 45.03 for additional information.

IAM360 STEP – ssssssss DDNAME – 'ddname' DATA SET MONITORED – '*dsname*'

Reason: Identifies the IAM job step name, the DDNAME and, optionally, the data set name of the IAM file being processed by the information service routines of the IAM monitor facility.

Action: None, information message only.

IAM361 INFO REQUESTED BY PROGRAM '*program*' PERFORMING '*description*' PROCESSING

Reason: Identifies the IAM processing program that requested the printing of the information block, the type of processing being performed, and the time the information block was printed by the information service routines of the IAM INFO report.

Action: None, information message only.

IAM362 DATA CHARACTERISTICS

Reason: Heading line which always precedes the IAM data characteristics when listed by the IAMINFO report.

Action: None, information message only, for further detail on these fields see Section 10.72: IAM RUN TIME STATISTICS.

IAM363 IAM FILE CHARACTERISTICS

Reason: Heading line which always precedes the IAM file characteristics when listed by the IAMINFO report.

Action: None, information message only, for further detail on these fields see Section 10.72: IAM RUN TIME STATISTICS.

IAM364 IAM OVERFLOW CHARACTERISTICS

Reason: Heading line which always precedes the IAM overflow characteristics when listed by the IAMINFO report.

Action: None, information message only, for further detail on these fields see Section 10.72: IAM RUN TIME STATISTICS.

80.03 CONTINUED . . .

IAM365 IAM EXECUTION STATISTICS

Reason: Heading line which always precedes the IAM execution statistics when listed by the IAMINFO report.

Action: None, information message only, for further detail on these fields see Section 10.72: IAM RUN TIME STATISTICS.

IAM366 IAM COMMAND EXECUTED SUMMARY

Reason: Heading line which always precedes the IAM command summary when listed by the IAMINFO report.

Action: None, information message only, for further detail on these fields see Section 10.72: IAM RUN TIME STATISTICS.

IAM367 THERE WAS INSUFFICIENT STORAGE AVAILABLE TO ACQUIRE ADDITIONAL BUFFERS

Reason: IAM's Real Time Tuning tried to acquire additional data buffers but there was insufficient virtual storage available.

Action: Increase the REGION value on the JOB statement or the EXEC statement so the next execution of the job will have storage available to acquire additional buffers.

IAM368 SPECIFYING A BUFNO VALUE GREATER THAN nn MAY IMPROVE PERFORMANCE

Reason: IAM's Real Time Tuning buffer management found that, for this mix of data and file processing commands, if additional buffers had been available they would have been acquired. Additional buffers were not acquired because it would have exceeded the maximum buffers allowed for this job.

Action: If you wish to increase the maximum buffers for this job, specify the MAXBUFNO=nn parameter on the IAM Override Control statement (Section 30.03) for this IAM file.

Note: If the number of I/O's (EXCPs) for this file is relatively small, there is no real need to increase the maximum number of buffers.

IAM371 INCREASING THE NUMBER OF PRIME EXTENSION BLOCKS MAY IMPROVE PERFORMANCE

Reason: An attempt to insert a record into prime extension failed because the prime extension was full. The record was inserted into independent overflow instead.

Action: Define the IAM file with a larger prime extension.

IAM372 IAM EXTENDED AREA CHARACTERISTICS

Reason: This heading line always precedes the description of the IAM Extended Area (Enhanced format files only).

IAM373 REORGANIZATION OF THIS FILE IS RECOMMENDED

Reason: During Open and/or Close processing, IAM noticed that the size of the extended overflow area was such that it could cause a performance degradation. As a preventative measure, it is recommended, but not required, that the file be reorganized to move records out of the overflow area. This IAMINFO message should be accompanied with an IAMW22 message, which will indicate the reason that a reorganization is being recommended.

Action: No action is required, but a file reorganization is recommended to be scheduled in the near future.

80.03 CONTINUED . . .

IAM400 *'processing function*' – '*progname*' – VER v.r. – INNOVATION DATA PROCESSING – DATE: '*yyddd*' PAGE: '*nn*'

Reason: Generalized page header for the named 'processing functions' which are provided by the 'program' named in the printed title.

Action: None, information message only.

IAM401 PARM DATA – * parm-field-data *

Reason: Displays the program control information specified in the "PARM=" field of the EXEC statement. This data will not be displayed if program is invoked under TSO.

Action: None, information only.

IAM402 INVALID CONTINUATION

Reason: User coded a delimiting comma following the last keyword on a control statement and did not provide the next logical record.

Action: Correct and resubmit job.

IAM403 REQUIRED OPERAND(S) NOT SPECIFIED – cc...cc

Reason: The operand cc...cc is required for the execution of the command. It must be specified; no defaults are available.

Action: Correct and resubmit job.

IAM404 WORKING STORAGE AREA SIZE OF nn BYTES EXCEEDED – SUBCOMMAND IGNORED

Reason: The maximum number of working storage bytes available to the subcommands has been exceeded.

Action: The number of bytes available for subcommand working storage is derived from the 'MAXCORE' operand, which has a lower limit of 1000 bytes and an upper limit of 120000 bytes. If the upper limit has been reached, call INNOVATION for technical support. If the upper limit has not been reached, resubmit the job specifying a value for 'MAXCORE' greater than the number displayed.

IAM405 MAXIMUM CONTINUATION COUNT OF nnnn EXCEEDED – COMMAND FLUSHED

Reason: The user control statement used too many continuations.

Action: Reduce the number of continuations to the value nnnn. Resubmit the job.

80.03 CONTINUED . . .

IAM407 CONTROL STATEMENT ERROR – '*action taken*'

Reason: An error was encountered during the processing of user supplied control statements. Always preceded by one or more messages which define and delimit the error(s). The 'action taken' is one of the following:

1. **JOB TERMINATED**—Processing will stop after the first error has been encountered.
2. **SKIPPING FOR COMMAND**—Processing will continue for all command statements within the SYSIN data set.
3. **RE-ENTER COMMAND OR END**—Message when the SYSIN data set is assigned to a TSO terminal. Re-enter command in error or 'END' to complete the processing.

Action: Correct and resubmit job.

IAM408 NO CONTROL STATEMENTS WERE FOUND JOB TERMINATED

Reason: SYSIN data set is empty or contained only comment statements ('*' in column 1).

IAM410 DSNAME='dsname' – '*error description*' – PROCESSING BYPASSED

Reason: The 'data set name' printed encountered the 'error' described and subsequent processing was bypassed. The error description can be one of the following:

- NOT RECOGNIZABLE IAM FILE
- TRKCALC FUNCTION FAILED
- NOT KSDS VSAM
- FILE HAS ALTERNATE INDEX
- NAMEDS DATA NOT RETURNED
- UNABLE TO LOCATE DATA/INDEX
- KEYLEN GREATER THAN 250 BYTES
- RKP GREATER THAN 4096
- FILE WAS NEVER LOADED
- IS NOT VSAM OR IAM

Action: Check the data set name for correctness. If the wrong data set name was specified, specify the correct data set name and resubmit the job. If the message indicates the data set is not an IAM file it is because the IAM control record was not present. Use program IAMRECVR specifying the IAM file characteristics on the 'RECOVER' subcommand to facilitate data recovery.

IAM413 DSN/DSG INDEX ERROR – '*error description*'

Reason: When using the index level option to select data sets or data set groups the user either:

1. Specified too many index levels. (The maximum is 22).
2. Failed to provide significant data. (The leading periods were followed by a comma or blank).

Action: Correct and resubmit.

80.03 CONTINUED . . .

IAM418 COMMAND PROCESSING DETECTED ERROR – '*action taken*'

Reason: An error was encountered during the processing of the subcommand specified on a user supplied control statement. Always preceded by one or more messages that define and delimit the error(s). The action taken is one of the following:

RE-ENTER COMMAND OR END – message for user when the SYSIN data set is assigned to a TSO terminal. Re-enter command in error or 'END' to complete the processing.

SKIPPING FOR COMMAND – processing will continue for all command statements found within the SYSIN data set.

REVERTING TO SYSIN – the error occurred when reading from an alternate command input source. Processing will continue for commands in the SYSIN data set.

Action: Correct and resubmit job.

IAM421 LOCATE ERROR – '*error description*' – DSN='*dsname*'

Reason: A LOCATE SVC was issued requesting identification of the component named by DSN='dsname'. The LOCATE either failed or returned a component:cluster name type code that is not currently supported.

If the 'error description' is in the form: RETURN CODE rc - REASON IGGOCLAA - 'reason number' The LOCATE failed. Error codes are documented in the IBM System Messages Manual; under message number IDC3009I.

If the 'error description' is in the form: ENTYPE – C(X'xx') The program encountered a type of component entry that it does not presently support.

If the 'error description' is in the form: NAMEDS – C(X'xx') The program encountered a type of cluster entry that it does not presently support.

If the 'error description' is in any other form, an error exists in the catalog.

Note: This message only appears if DFEFERRPRT=YES had been setup earlier with IAMZAPOP or specified on the control statement. This is only a warning message. The program does not associate the component shown by DSN='dsname' with a cluster name, processing continues.

Action: Do one of the following:

For the RETURN CODE type of error, look up the codes under message IDC3009I. The most likely cause is a STEPCAT or JOBCAT DD statement was not supplied for a user catalog that has entries in it for components on the volume(s) being processed, and those entries are not ALIAS'ed.

For any other type of errors, contact INNOVATION technical support for assistance.

IAM465 UNABLE TO OPEN (DDNAME=DSN=)'*name*' – '*reason*'

Reason: The 'ddname' named in the message failed to OPEN for the 'reason' shown in the text. The recognized reasons are:

ABEND exit taken.

VOL='volser'

denied by OPEN exit.

DD='ddname' statement missing/ misspelled or incorrectly specified.

RC=nn ERROR CODE=nnn

Action: If the named 'ddname' is required for the requested operation correct the error and resubmit the job, otherwise the message may be ignored.

80.03 CONTINUED . . .

IAM471 DDNAME='*ddname*' I/O ERROR – SYNAD='*message*'

Reason: A permanent I/O error was detected on the data set referenced by the DDNAME 'ddname'.

Action: Examine the SYNADAF message to determine the cause of the error. The format of this message is documented in IBM SRL publications. Call INNOVATION for additional assistance.

IAM476 DDNAME='*ddname*' REFERENCES A DEVICE TYPE OTHER THAN DISK

Reason: The DDNAME 'ddname' listed can only be allocated to a disk device.

Action: Check the unit specification in the JCL for errors, correct and resubmit.

IAM484 INTERNAL LOGIC ERROR – JOB TERMINATED

Reason: The program has encountered an illogical condition.

Action: Call INNOVATION for additional assistance AFTER obtaining a storage dump.

IAM485 SORT FAILURE HAS OCCURRED RC='nn' – '*action description*'

Reason: Your installation's sort product has set a return code other than zero. The return code problem description can be found in the sort manual supplied with the sort program product. In many cases the sort program will provide an error message on DDNAME SYSOUT and or on the system console.

Action: If you cannot correct the problem from this information, call INNOVATION for additional assistance.

IAM491 '*function*' FUNCTION STARTED TIME=hh.mm.ss

Reason: Identifies type of function and time the function started.

Action: None, information message only.

IAM492 '*function*' FUNCTION ENDED – hh.mm.ss CONDITION CODE – nnnn

Reason: Identifies type of function and time the function ended. The return code is printed if it is non-zero.

Action: Review the completion code. If the completion code is greater than zero, check the output for the preceding messages that describe the reason for a non-zero completion.

IAM493 '*function*' – BYPASSED – PRIOR FUNCTION TERMINATED WITH KEYWORD OR COMMAND DETECTED ERROR

Reason: A previous command upon which this function may be dependent encountered serious errors.

80.03 CONTINUED . . .

IAM495 NO RECORDS MATCHED SELECTION CRITERIA

Reason: The selection criteria specified did not cause any records to be selected for processing. If program IAMSMFVS, this message will also appear if the required SMF record types are not being collected.

Action: Make sure the selection criteria is correct.

If IAMSMFVS: Make sure you are collecting SMF records types 4 or 30 subtype 4 and type 64. If you are running against an SMF history tape, make sure that the required record types are being copied to the history tapes.

If IAMSMF: Make sure you are collecting SMF records types 4 or 30 subtype 4 and type 14,15, 64. If you are running against an SMF history tape, make sure that the required record types are being copied to the history tapes.

IAM496 MODULE '*modname*' – NOT USABLE WITH RELEASE v.r. PROGRAMS – EXECUTION TERMINATED

Reason: Module 'modname' is a release/version that is incompatible with the load module being executed.

Action: Check for a STEPLIB/JOBLIB DD statement pointing to a library other than the correct library for the product level you expect to use. If present, correct the library name and resubmit the job. If there is no STEPLIB/JOBLIB present, the module 'modname' is being obtained from a LINKLIST library. Add a STEPLIB or JOBLIB DD statement specifying the correct load module library and resubmit the job.

IAM497 cc..cc ABNORMALLY TERMINATED DUE TO KEYWORD/COMMAND DETECTED ERRORS

Reason: The common parsing routine encountered errors in parsing the user specified control statements. In addition, the user has set the option KWDCC=ABEND via program IAMZAPOP. Always preceded by one or more error description messages.

Action: Previous message(s) describe the error(s); see those messages for further details.

IAM498 cc...cc PROCESSING COMPLETED WITH ERRORS

Reason: The named program completed the requested processing but encountered abnormalities in the process.

Action: Check the output for preceding messages that may describe the errors in detail.

IAM499 cc...cc PROCESSING COMPLETED

Reason: The named program has completed processing as requested.

Action: Check the output for messages that describe the results.

IAM516 CAMLST REGISTERS R0=nnnnnnnn R1=nnnnnnnn R15=nnnnnnnn

Reason: A CAMLST request failed. The type of CAMLST function (shown in the immediately preceding message) and the registers make diagnosing the problem relatively simple. The return code(s) from catalog management is(are) documented in the IBM SRL SYSTEM PROGRAMMING LIBRARY: DATA MANAGEMENT (for MVS) or CATALOG ADMINISTRATION GUIDE (for MVS/XA) or DATA MANAGEMENT FOR SYSTEM PROGRAMMERS (for non-MVS).

Action: Check the return code(s) and take corrective action if the error is apparent or call INNOVATION for additional assistance.

80.03 CONTINUED . . .

IAM530 MODULE '*modname*' NOT FOUND – '*ddname*' – '*dsname*'

Reason: A BLDL was issued for the module 'modname' in the dataset 'dsname' referenced by 'ddname' and the module was not found. The module is required to support a requested processing function.

Action: Verify that the library dataset name 'dsname' specified on the 'ddname' statement was correct. If incorrect, correct it and resubmit the job. If the dataset name is correct, check the IAM product install listing to see that all steps executed successfully. If required, rerun the installation steps missed. If necessary contact INNOVATION for additional assistance.

IAM531 MODULE '*modname*' CONTAINS NO TEXT RECORDS – '*ddname*' – '*dsname*'

Reason: When an attempt was made to read the module 'modname' in the dataset 'dsname' referenced by 'ddname' and no text records were found. The module is required to support a requested processing function.

Action: Verify that the library dataset name 'dsname' specified on the 'ddname' statement was the correct one. If incorrect, change and resubmit the job. If the dataset name is correct, check the IAM product install listing to see that all steps executed successfully. If required, rerun the installation steps missed. If necessary contact INNOVATION for additional assistance.

IAM532 MODULE '*modname*' I/O ERROR READING – '*ddname*' – '*dsname*'

Reason: When an attempt was made to read the module 'modname' in the dataset 'dsname' referenced by 'ddname' an I/O error was encountered. The module is required to support a requested processing function.

Action: Check the SYNAD error message(s) and MINI DUMP produced to see if the cause of the error is an obvious one. If so, correct and resubmit the job. If necessary contact INNOVATION for additional assistance.

IAM533 MODULE '*modname*' I/O ERROR WRITING – '*ddname*' – '*dsname*'

Reason: When an attempt was made to write the module 'modname' in the dataset 'dsname' referenced by 'ddname' an I/O error was encountered.

Action: This is a serious error which may result in the load module library referenced by 'dsname' being unusable. If necessary, restore or recreate the library. Check the SYNAD error message(s) and MINI DUMP produced to see if the cause of the error is an obvious one. If so, correct and resubmit the job. If necessary contact INNOVATION for additional assistance.

IAM534 '*ddname*' OPEN FAILURE – AVAILABLE COMMANDS: HELP, END

Reason: The required 'ddname' statement could not be OPENed. Commands which access modules in the 'ddname' dataset cannot be executed.

Action: If the 'ddname' dataset is required for the operation being attempted, correct and resubmit the job. Otherwise, none, information only.

IAM535 MODULE '*modname*' READ UNSUCCESSFUL – '*command*' BYPASSED

Reason: The user requested function 'command' be performed against module 'modname' but the module was not successfully read. This message will be preceded by message IAM530, IAM531, or IAM532 detailing the reason the read failed.

Action: See message IAM530, IAM531, or IAM532, as required.

80.03 CONTINUED . . .

IAM538 MODULE '*modname*' – WRONG VER/LEVEL '*ddname*' – '*dsname*'

Reason: Module 'modname' is a release/version that is incompatible with the load module being executed.

Action: Verify that the 'ddname' DD statement is pointing to the correct library for the product level you expect to use. If incorrect, change and resubmit the job. If the data set name is correct, check the IAM product install listing to see that all steps were executed successfully. If required, rerun the installation steps missed. If necessary, contact INNOVATION for additional assistance.

IAM539 MODULE '*modname*' '*function*' – '*ddname*'– '*dsname*'

Reason: The module 'modname' from the dataset dsname' referenced by 'ddname' has had function 'function' performed as requested by the user.

Action: None, information only.

IAM540 cccccccc CONTAINS INVALID CHARACTERS – ZAP REJECTED

Reason: The value specified for the operand 'cccccccc' contained one or more characters that were not A-Z, 0-9, $#@.

Action: Remove the invalid character(s) and resubmit the job.

IAM541 THE FOLLOWING OPTIONS HAVE BEEN CHANGED IN – '*dsname*'

Reason: The AUDIT function has found changed options in the options table. The changed options will be listed.

Action: N/A

IAM542 cccccccc CONTAINS INVALID INDEX STRUCTURE – ZAP REJECTED

Reason: The value specified for the operand cccccccc contains two (2) or more consecutive periods (..) in violation of operating system dataset naming conventions.

Action: Correct the error(s) and resubmit the job.

IAM543 cccccccc DOES NOT CONTAIN CHARACTER STRING cccccccc – ZAP REJECTED

Reason: The value specified for the operand cccccccc does not contain the character string cccccccc as the required start of an index level.

Action: Correct the error(s) and resubmit the job.

IAM544 cccccccc REJECTED – EXCEEDS MODIFIABLE PORTION OF IAMOPT

Reason: The length of the value specified for the operand cccccccc taken in conjunction with the offset operand exceeds the modifiable portion of the IAMOPT table.

Action: Correct the error(s) and resubmit the job.

80.03 CONTINUED . . .

IAM545 VERIFY FAILED – CHAR/HEX PRINT FORCED

Reason: The VERIFY of existing contents failed. A character/hexadecimal print of the module IAMOPT is produced. Always preceded by message IAM544.

Action: See message IAM544.

IAM546 AUDIT REQUEST COMPLETE ALL DEFAULTS SET IN – '*dsname*'

Reason: The AUDIT function has found no changed options in the options table.

Action: N/A

IAM547 AUDIT REQUEST COMPLETE FOR – '*dsname*'

Reason: The AUDIT function has completed. Changes were found and listed.

Action: N/A

IAM550 ERRORS ENCOUNTERED DURING EXECUTION – REWRITE CANCELED

Reason: One or more modules from the SYSLIB data set were scheduled for rewrite at termination or by the execution of a REWRITE command. However, previous commands failed to complete successfully. The errors encountered have been documented by preceding error messages.

Action: Correct the error conditions documented by the preceding error messages and resubmit the job.

IAM561 NON-NUMERIC DATA VALUE SPECIFIED FOR THE [FROMDATE | TODATE] KEYWORD

Reason: An invalid character string was specified for the indicated keyword.

Action: Correct the error condition. For information on valid values, refer to the IAMSMFVS section of the manual.

IAM562 INVALID VALUE SPECIFIED FOR [FROMDATE | TODATE]

Reason: An invalid numeric string was specified for the indicated keyword.

Action: Correct the error condition. For information on valid values, refer to the IAMSMFVS section of the manual.

IAM569 WARNING – cccccccc NOT CATALOGED – DSN='*dsname*'

Reason: The dataset 'dsname' indicated is the new default for the file type indicated by cccccccc. A CAMLST LOCATE was issued to verify that the dataset is cataloged as required for IAM to dynamically allocate. The data set name was not found by LOCATE. Message IAM516 detailing the CAMLST return code(s) is always printed following this message.

Action: See message IAM516. This is only a warning message. The data set name listed has become the new name of the indicated default.

80.03 CONTINUED . . .

IAM574 'request' – DDNAME='ddname' – 'action'

Reason: Failed I/O 'request' resulting in the named 'action':

- ** DD NOT OPEN–EXTRACT REQUEST DENIED
- GET FAILED—EOF FORCED
- CLOSE FAILED—JOB TERMINATED
- OPEN FAILED—COPY REQUEST DENIED
- OPEN FAILED—EXTRACT REQUEST DENIED
- OPEN FAILED—REPORT REQUEST DENIED
- ** OPEN FAILED—REQUEST DENIED
- OPEN FAILED—TO DDNAME COPY DENIED
- RJFCB FAILED—REQUEST DENIED

The 'ddname' shown in the message is required as either INPUT TO or OUTPUT FROM a processing program. An unexpected failing request for an I/O related service against that DDNAME has forced the program to take the remedial action shown in the message text.

Action: Correct the reason for the error and resubmit the job.

IAM600 MAXSTACK VALUE OF nnnnnn EXCEEDED – SMF RCD PROCESSING TERMINATED

Reason: The SMF record processing subcommand being executed utilizes an in storage stack specified by the operand MAXSTACK. This storage stack is full. The specified size is insufficient to allow all of the selected SMF records to be processed. IAMSMF terminates without producing a report.

Action: Re-execute the program and specify the operand MAXSTACK= with a value greater than the value nnnnnn printed in this message.

IAM601 SMF RECORDS – READ.nnnnnnnn USED.nnnnnnnn DROPPED.nnnnnnnn

Reason: Documents the SMF records processed as follows:

READ – number of SMF records read from the input data set. May not reflect the total number of records in the data set if an IAM600 message was issued.

USED – number of SMF records selected from the input data set by the criteria specified by the user.

DROPPED– number of SMF records dropped from the input data set because of length checking, or other user specified criteria.

Action: None, information only.

IAM602 FOLLOWING SMF RECORD DROPPED – LENGTH CHECK

Reason: User specified the CHECKLENGTH operand, causing all SMF records selected to be checked against a table of minimum lengths. This record failed the length check. The first 32 bytes of the record are printed in hexadecimal.

Action: Use the keyword RECSIZE=nn on the report control statement to specify an appropriate size.

80.03 CONTINUED . . .

IAM603 VSAM ERROR – (DDNAME=DSN=)'*name***' '***error description***' – R15=xxxxxxxx – CODE=xxxxxxxx**

Reason: An error occurred during the processing of a VSAM data set. The error description will be one of the following:

- CLOSE failed—close of VSAM data set.
- GENCB failed—generation of a control block.
- GET failed—get next record.
- OPEN failed—OPEN of VSAM data set.

Action: Check the values of R15 and CODE against the return codes listed in the 'VSAM REFERENCE FOR MVS/370 DFP', 'VSAM ADMINISTRATION: MACRO INSTRUCTION REFERENCE FOR MVS/XA', or 'OS/VS VIRTUAL STORAGE ACCESS METHOD PROGRAMMER'S GUIDE', to determine the cause of the error. If possible, correct and re-execute the job. Call INNOVATION for additional assistance.

IAM619 MODULE '*modname***' TOO LARGE TO PROCESS – '***ddname***' – '***dsname***'**

Reason: IAMZAPOP attempted to read the module 'modname' in the dataset 'dsname' referenced by the statement 'ddname'. The module was required to support a processing functions but, due to the size of the module (or previously read modules), not enough buffer storage was available to complete the read.

Action: Run IAMZAPOP specifying 'ZAP BUFSIZE=nnnn', where nnnn is the size of the buffer in bytes. Specify a value between 122,880 and 3,145,728. After the buffer size has been changed, rerun the job that failed.

IAM620 MAXJOB VALUE OF nnnnn EXCEEDED – ADDITIONAL JOB NAMES BYPASSED

Reason: The maximum number of unique job names tabled by IAMSMFVS has been exceeded. SMF records that match the specified selection criteria but with job names other than those already tabled will be bypassed and not reported.

Action: Specify a MAXJOB value greater than nnnnn but less than or equal to 32,000 and, if required, rerun the job.

IAM621 MAXDSN VALUE OF nnnn EXCEEDED – ADDITIONAL DATA SET NAMES BYPASSED

Reason: The maximum number of unique dataset names tabled by has been exceeded. Records that match the specified selection criteria but with dataset names other than those already tabled will be bypassed and not reported or will be shown in a second report if present.

Action: Specify a MAXDSN value greater than nnnnn but less than or equal to 32,000 and, if required, rerun the job.

IAM622 MAXDSN VALUE OF nnnnn EXCEEDED – ADDITIONAL VSAM CLUSTER RELATIONSHIPS BYPASSED

Reason: IAMSMFVS issues a LOCATE for each ICF/VSAM DATA and/or INDEX component and adds the resulting cluster name to the tabled data set names. Those DATA and/or INDEX components that remain unrelated (I.E.: have not had the cluster name appended) when the value of MAXDSN is exceeded will be printed as unique data sets.

Action: ICF/VSAM clusters can take up to three (3) entries in the dataset name table that is generated from the MAXDSN operand. If you only plan to extract information on ICF/VSAM clusters and you expect to have approximately 1000 clusters listed, specify a MAXDSN no less than 3000. If the error reoccurs, specify a MAXDSN value greater than nnnnn but less than or equal to 32,000 and, if required, rerun the job.

80.03 CONTINUED . . .

IAM634 SEQUENCE CHECKS FORCE SORTING OF DATA BEFORE IAM CREATION

Reason: The file recovery program found the records it was processing were not in ascending sequential order. An IAM file creation expects the IAM file to be loaded in order. The set of keys the recovery program is processing can not be used as is to load an IAM file. The Independent Overflow records in an IAM file may not be in collating sequence. Recovery of a file with Independent Overflow could result in this message.

Action: The recovered records must be sorted before they can be used to load an IAM file.

IAM635 LOG AND IAM FILE ARE INCOMPATIBLE

Reason: The file recovery program found the log file records it was applying were not compatible with the IAM file it was rebuilding.

Action: The log file being applied and the IAM file in question should be checked to see if they actually represent the same data before attempting to continue with a file recovery.

IAM636 APPLY FAILED – RECORD KEY FOLLOWS –

Reason: The file recovery program found the record it was applying was not compatible with the IAM file it was rebuilding.

Action: The log file being applied and the IAM file in question should be checked to see if they actually represent the same data before attempting to continue with a file recovery.

IAM637 IAM INTERNAL BLOCK –

Reason: The block shown contains file control information.

Action: None, information message only.

IAM638 IAM BLOCK NUMBER nnnnnn –

Reason: The block shown is relative block number 'nnnnnn' in the file.

Action: None, information message only.

IAM700 Innovation Access Method Trace VER nnnnnnnn Date:yyyy.ddd Page:nnn

IAM701 DDNAME:ddname DSN:dataset name

IAM702 Type Time Request RPL/RBN PLH OPTCD RC/RINFO RECLEN Key

IAM703 ---- ---- ------- ------- --- ----- -------- ------ ---

Reason: IAM700 - IAM703 are headings for IAM trace output.

IAM704 Trace Ended

Reason: indicates the end of IAM tracing.

80.03 CONTINUED . . .

IAM705 There were trace lock failures

Reason: If any trace lock failures occurred during processing, this message will be issued.

IAM710 IOS 08.33.38.1900 PUT 00023A10 00040298 2040 01FE 00002D00

Reason: This is the I/O Start trace detail line. The contents of each field are as follows:

- Type—IOS Request—One of the following, depending on the type of
- I/O requested: GET, PUT, POINT, ERASE, CHECK or ENDREQ.
- Time—Time the trace record was generated.
- RPL/RBN—The address of the RPL used to request the I/O.
- PLH—The address of the active PLH for this I/O.
- OPTCD—The values of RPLOPTCD1 and RPLOPTCD2 at the time of the request.
- RC/RINFO—blank
- RECLEN—The requested record length (if applicable).
- Key—The records RBA or key (if applicable).

Notes: When trace is activated on a PATH, the trace output is modified to include a suffix to the message number. When the trace entry is for a path, the suffix is "P", for the alternate index the suffix is "X", and for the base cluster, the suffix is "B"

IAM711 IOE 08.33.39.2200 PUT 00023A10 00040298 0000 01FE 0005A000

Reason: This is the I/O End trace detail line. The contents of each field are as follows:

- Type—IOE Request—One of the following, depending on the type of I/O requested: GET, PUT, POINT, ERASE, CHECK or ENDREQ.
- Time—Time the trace record was generated.
- RPL/RBN—The address of the RPL used for this request.
- PLH—The address of the active PLH for this I/O.
- OPTCD–N/A
- RC/RINFO–The return code from the I/O.
- RECLEN–The record length (if applicable).
- Key–The records RBA or key (if applicable)

Notes: When trace is activated on a PATH, the trace output is modified to include a suffix to the message number. When the trace entry is for a path, the suffix is "P", for the alternate index the suffix is "X", and for the base cluster, the ssuffix is "B"

80.03 CONTINUED . . .

IAM712 BFR 08.33.39.2300 WRITE 00000026 00040298 1008 0005A1FE

Reason: This is the Buffer Manager trace detail line. The contents of each field are as follows:

- Type—BFR Request—One of the following, depending on the type of processing required: READ, WRITE, RELEASE, FLUSH.
- Time—Time the trace record was generated.
- RPL/RBN—The Relative Block Number (RBN) of the requested data block.
- PLH—The address of the active PLH for this I/O.
- OPTCD–PLH option bytes 1/2.
- RC/RINFO–'OV' if record was from/to Independent Overflow.
- RECLEN–N/A.
- Key–The records RBA or key (if applicable).

IAM713 EXCP 08.33.39.2400 READ 00000024 00040298 0001

Reason: This is the EXCP trace detail line. The contents of each field are as follows:

- TYPE—EXCP Request—One of the following, depending on the type of processing required:
- PRFMT—Preformat additional blocks
- WREOF—Write EOF
- WRADD—Write new block
- READ—Read single or multiple blocks
- WRITE—Write single or multiple blocks
- Time—Time the trace record was generated.
- RPL/RBN—Relative Block Number (RBN) of first block processed.
- PLH—The address of the active PLH for this I/O.
- OPTCD—Number of blocks to process.
- RC/RINFO—N/A.
- RECLEN—N/A.
- Key—N/A.

80.03 CONTINUED . . .

IAM714 XTND 08.34.02.0100 EXTEND 00000028 00000000 0102 0000 20980001

Reason: This is the XTND trace detail line. The contents of each field are as follows:

- TYPE—XTND Request—One of the following, depending on the type of EXTEND processing required:
- EXPAND—Expand into allocated but unused area.
- EXTEND—Obtain new extent.
- EXP/EXT—Expand and obtain new extent.
- Time—Time the trace record was generated.
- RPL/RBN—RBN of first new block after EXTEND.
- PLH—Number of new index blocks after EXTEND.
- OPTCD—Volume number and extent number of new extent.
- RC/RINFO—Return code from EXTEND processing.
- RECLEN—N/A.
- Key—EXTEND processing internal flags.

IAM715 hh.mm.ss.nnnn PUT xxxxxxxx oooooooo rrrrrrrr llllllll kkkkkkkkkkkkkkkkkkkk

Reason: This is the file load detail trace line. The contents of each field are as follows:

- **hh.mm.ss.nnnn** – Time the request was issued.
- **PUT** – VSAM I/O request type. Normally for file load, this will only be a PUT. Other types of I/O requests (i.e., GET) are invalid during a file load, although if such a request is attempted it will be included on the trace.
- **xxxxxxxx** – The address of the Request Parameter List (RPL), in hexadecimal.
- **oooooooo** – The 4 bytes, in hexadecimal, of the RPL OPTCD field.
- **rrrrrrrr** – The return code and logical error code, in hexadecimal, from the RPLFDBWD field.
- **llllllll**—The length of the record being written to the file, from the RPLRLEN field.
- **kkkkkk…** – The key of the record being written. For ESDS files, this will be the RBA of the record generated by IAM. The key will be displayed in either hexadecimal, or in character, depending on the name of the TRACE DD statement. The key will be continued on subsequent lines, if necessary.

IAM716 +kkkkkkkk

Reason: This is the continuation of the key of the record being processed, from the preceding IAM715 message.

80.04 IAM JOURNAL EXIT WTO MESSAGES

IAMJ01 IAMDD ddname: INITQ BROKEN, JOURNAL PROCESSING TERMINATED.

Reason: An error occurred during the IAM Journal Exit initialization processing. This error can occur when multiple concurrent I/O requests are active and are requiring service from the IAM Journal Exit while it is attempting to allocate and open the log data set.

Action: I/O requests will continue to be serviced to the IAM data set, however journalling is no longer being done by IAM for the indicated IAM data set. Contact Innovation Technical Support for assistance.

IAMJ02 IAMDD ddname: ALLOCATION OF LOG FILE FAILED, JOURNAL PROCESSING TERMINATED.

Reason: The dynamic allocation of the log file for the specified IAM data set has failed.

Action: I/O requests will continue to be serviced for the indicated IAM data set, however journalling is not being done. If you are having difficulty determining why the LOG data set is not able to be allocated, contact Innovation for assistance.

IAMJ03 IAMDD ddname: GETMAIN1 FAILED, JOURNAL PROCESSING TERMINATED.

Reason: There is insufficient below the 16 megabyte line storage available in the region for IAM to acquire the storage required to handle the requested journalling. The amount of storage being requested that resulted in this failure is generally less than 2K. The OPEN of the indicated IAM data set will be failed.

Return Codes: A reason code of 188(x'BC') is stored in the ACB error flags field (ACBERFLG) and the open request is failed with a return code of 8.

Action: Generally increasing the REGION parameter for the job step should correct this problem. If it does not, then contact Innovation Technical Support for assistance. This problem can be circumvented by turning journalling off for this file, through the use of the IAM overrides. (i.e., JRNAD=NONE).

IAMJ04 IAMDD ddname: OPEN OF LOG FILE FAILED, JOURNAL PROCESSING TERMINATED.

Reason: The OPEN of the log file has failed. There should be some IBM messages indicating the cause of the failure.

Action: I/O requests will continue to be serviced for the indicated IAM data set, however journalling is not being done. Correct the error condition based on the information available.

IAMJ05 IAMDD ddname: BUFFER GETMAIN FAILED, JOURNAL PROCESSING TERMINATED.

Reason: There is insufficient storage available to obtain the necessary I/O buffers for the IAM journal processing. For systems that are at a high enough level of DFSMS, the buffers are requested above the 16 megabyte line, otherwise the storage is requested from below the 16 megabyte line.

Action: I/O requests will continue to be serviced for the indicated IAM data set, however journalling is not being done. The journal exit requests storage for five (5) buffers, so multiply the block size of the data set by 5 to determine storage requirements, and adjust your region parameter as necessary.

IAMJ06 IAMDD ddname: SYNAD MESSAGE GOES HERE.......................

Reason: An I/O error has occurred on the IAM Log dataset. Review the I/O error cause from the message displayed.

Action: I/O requests will continue to be serviced for the indicated IAM data set, however journalling is not being done. Correct the problem causing the I/O error.

80.04 CONTINUED . . .

IAMJ08 IAMDD ddname: LOCATE FAILED FOR THE LOG FILE, JOURNAL PROCESSING TERMINATED.

Reason: In an attempt to prevent allocation and open errors, the IAM Journal exit program will verify that the required log file is in the catalog. If it is not found in the catalog, then the OPEN request for the IAM data set will also fail.

Return Codes: A reason code of 188(x'BC') is stored in the ACB error flags field (ACBERFLG) and the open request is failed with a return code of 8.

Action: Action: The OPEN request for the IAM file is failed. Either correct the error condition, or disable IAM journalling for this file and job step through the ACCESS override keyword JRNAD=NONE.

IAMJ09 IAMDD ddname: OBTAIN FAILED FOR THE LOG FILE, JOURNAL PROCESSING TERMINATED.

Reason: The IAM journal exit will attempt to make sure that the required log data set is allocated on the volume indicated by the catalog prior to attempting an allocation and open of this critical file.. If the required log data set is not found, then the OPEN for the IAM data set will be failed.

Return Codes: A reason code of 188(x'BC') is stored in the ACB error flags field (ACBERFLG) and the open request is failed with a return code of 8.

Action: Make sure that the required log data set is properly allocated and cataloged. Specifying the IAM ACCESS override JRNAD=NONE can be used to turn off journalling until the cause of the error is found.

IAMJ10 IAMDD ddname: DCB ATTRIBUTES OF LOG FILE ARE INCOMPATIBLE, JOURNAL PROCESSING TERMINATED.

Reason: The IAM journal exit will make sure that the attributes (RECFM, LRECL, and BLKSIZE) are appropriate as needed by the file being logged. If they do not meet the required criteria, the OPEN of the IAM data set will fail.

Return Codes: A reason code of 188(x'BC') is stored in the ACB error flags field (ACBERFLG) and the open request is failed with a return code of 8.

Action: Review the documentation on the IAM journalling exit to make sure that the DCB attributes selected for the LOG data set are appropriate, and change them as necessary. It is best to let IAM determine the DCB attributes, by just allocating the DASD space required for the LOG data set without specifying any DCB attributes.

IAMJ11 IAMDD dname: SAVEAREA GETMAIN FAILED, JOURNAL PROCESSING TERMINATED.

Reason: There is insufficient virtual storage available for the IAM Journal Exit. The OPEN of the IAM data set will be failed. The amount of storage being requested is generally less than 1K, and it can reside in either above or below the 16 megabyte line.

Return Codes: A reason code of 188(x'BC') is stored in the ACB error flags field (ACBERFLG) and the open request is failed with a return code of 8.

Action: Increase the REGION parameter of the job step that is failing with this error.

80.05 IAM RLS MESSAGES

IAML0001 – IAML0050 IAMRLS MESSAGES

IAML0001 LOGGING OF MESSAGES DISABLED

Reason: IAMRLS was unable to obtain storage for the message table used by IAMLOGER and has disabled logging of messages to RLSLOGDD.

Action: Increase the REGION size of the IAMRLS address space.

IAML0002 GLOBAL NAME TOKEN AREA RETRIEVED

Reason: IAMRLS was able to retrieve and reuse the Global Token area left from a previous iteration of the IAMRLS address space.

Action: None

IAML0003 UNABLE TO START VIF

Reason: IAMRLS received a non-zero return code from IAMSTART while attempting to start VIF.

Action: Check for other messages issued by IAMSTART to the SYSLOG which will provide further information as to the reason VIF could not start.

IAML0004 GETMAIN OR LOCK OBTAIN FAILURE

Reason: IAMRLS could not obtain the local lock needed for a CSA getmain, or there was not enough CSA available to satisfy the CSA getmain request.

Action: Contact Innovation for help in diagnosing the cause of this problem.

IAML0005 GLOBAL NAME TOKEN AREA CREATE FAILED

Reason: IAMRLS was unable to create a Global Token area.

Action: Contact Innovation for help in diagnosing the cause of this problem.

IAML0006 DATASPACE BUILD FAILED

Reason: IAMRLS was unable to create a dataspace.

Action: Contact Innovation for help in diagnosing the cause of this problem.

IAML0007 SYSEVENT FAILED

Reason: The SYSEVENT macro issued to make the IAMRLS address space non-swappable failed.

Action: Contact Innovation for help in diagnosing the cause of this problem.

IAML0008 LOAD OF SERVICE MODULE FAILED

Reason: IAMRLS was unable to load one of it's service modules.

Action: Make sure you have an IAM load library that has all of the modules distributed on the IAM distribution tape included in either the LINKLIST or a JOBLIB/STEPLIB.

80.05 CONTINUED . . .

IAML0009 LOCAL NAME TOKEN AREA CREATE FAILED

Reason: IAMRLS was unable to create the Local Name Token area.

Action: Contact Innovation for help in diagnosing the cause of this problem.

IAML0010 GETMAIN OF CELL POOL STORAGE FAILED

Reason: IAMRLS was unable to getmain storage for the cell pools set up during initialization.

Action: Raise the region size for the IAMRLS address space.

IAML0011 TASK WORK AREA GETMAIN FAILED

Reason: The getmain for the TASK workarea storage failed.

Action: Raise the region size for the IAMRLS address space.

IAML0012 ATTACH PROCESSOR FAILED, CANNOT CONTINUE

Reason: IAMRLS was unable to attach one of it's subtasks.

Action: Contact Innovation for help in diagnosing the cause of this problem.

IAML0013 GLOBAL NAME TOKEN AREA DELETED

Reason: IAMRLS deleted the Global Token Area during shutdown.

Action: None.

IAML0014 OPEN/CLOSE TASK ATTACHED

Reason: IAMRLS attached the OPEN/CLOSE subtask.

Action: None.

IAML0015 LOGGING TASK ATTACHED

Reason: IAMRLS attached the LOGGING subtask.

Action: None.

IAML0016 WORKLOAD MANAGER ATTACHED

Reason: IAMRLS attached the WORKLOAD MANAGER subtask.

Action: None.

IAML0017 JOURNAL TASK ATTACHED

Reason: IAMRLS attached the JOURNAL subtask.

Action: None.

80.05 CONTINUED . . .

IAML0018 TRACE TASK ATTACHED

Reason: IAMRLS attached the TRACE subtask.

Action: None.

IAML0019 IAMRLS ESTAE ENTERED FOR ABEND SXXX

Reason: The ESTAE for module IAMRLS was entered for an abend condition.

Action: Contact Innovation for help in diagnosing the cause of this problem.

IAML0020 PRELOAD OF PROCESSING MODULE FAILED

Reason: IAMRLS was unable to preload some of the processing modules.

Action: Check the IAM load library to make sure it contains all of the modules distributed with the IAM product.

IAML0021 ADDRESS SPACE ALREADY ACTIVE

Reason: IAMRLS detected an address space running under the same name.

Action: None.

IAML0022 IAMBREST SUBTASK ATTACHED

Reason: IAMRLS attached the DYNAMIC BACKOUT subtask.

Action: None.

IAML0049 ADDRESS SPACE STILL ACTIVE, ISSUE STOP, QUIESCE, OR QUIESCE FORCE TO DEACTIVATE.

Reason: The IAMRLSX program was executed to delete the Global Token Area from CSA storage, but a flag was set in the token area indicating that IAMRLS was still active.

Action: This program should not be run until the IAMRLS address space has been deactivated. Quiesce IAMRLS and run this program again.

IAML0050 GLOBAL NAME TOKEN AREA DELETED

Reason: IAMRLSX successfully deleted the Global Token Area from CSA storage.

Action: None

IAML0051 – IAML0075 IAMOPCOM MESSAGES

IAML0051 IAMOPCOM WAITING

Reason: IAMOPCOM, the operator communications task for IAMRLS, is waiting for work.

Action: None

80.05 CONTINUED . . .

IAML0052 IAMOPCOM SHUTTING DOWN

Reason: IAMOPCOM has ended operator communications and is shutting down.

Action: None

IAML0053 IAMOPCOM MODIFY RECEIVED

Reason: IAMOPCOM has received a modify command.

Action: None

IAML0054 IAMOPCOM RECEIVED INVALID COMMAND

Reason: IAMOPCOM has received an invalid command.

Action: Check the spelling and syntax of the command entered.

IAML0055 IAMOPCOM RECEIVED INVALID MODIFY

Reason: IAMOPCOM has received an invalid modify command.

Action: Check the spelling and syntax of the modify command entered.

IAML0056 TRACE COMMAND COULD NOT BE PROCESSED

Reason: IAMOPCOM was unable to load the trace processing program.

Action: Check the contents of the IAM load library to ensure all of the distributed modules are present.

IAML0057 JSWITCH COMMAND COULD NOT BE PROCESSED

Reason: The JSWITCH command was entered to switch the journal datasets but IAMRLS is using the SYSTEM LOGGER for the journal so the switch could not be processed.

Action: None

IAML0058 IAMOPCOM LOAD OF IAMBSTAE FAILED, ESTAE NOT ACTIVE

Reason: IAMOPCOM was unable to load the IAMBSTAE module.

Action: Check the contents of the IAM load library to ensure all of the distributed modules are present.

IAML0059 IAMOPCOM ESTAE ACTIVATION FAILED

Reason: IAMOPCOM was unable to activate it's estae routine.

Action: Contact Innovation for help in diagnosing the cause of this problem.

IAML0060 IAMOPCOM COMMAND INVALID, MISMATCHED QUOTES

Reason: On an APPLY or RESTORE command, the dataset name entered contains mismatched quotes.

Action: Re-enter the command correctly.

80.05 CONTINUED . . .

IAML0061 IAMOPCOM COMMAND INVALID, MISMATCHED PARENTHESIS

Reason: On an APPLY or RESTORE command, the dataset name entered contains mismatched parenthesis.

Action: Re-enter the command correctly.

IAML0062 MESSAGES COMMAND INVALID

Reason: An invalid MESSAGES command was entered.

Action: Check the syntax of the command and re-enter it correctly.

IAML0063 DISPLAY COMMAND INVALID

Reason: An invalid DISPLAY command was entered.

Action: Check the syntax of the command and re-enter it correctly.

IAML0064 SELECT TABLE DATASET LIST

Reason: This is the header message for the list of SELECT TABLE datasets displayed as a result of the DISPLAY,SELECTTB command.

Action: None

IAML0065 DSN: dsname or NO MATCHING DATASETS FOUND

Reason: This message is written once for every dataset in the Select Table that matches the selection criteria or once if no datasets are found.

Action: None

IAML0066 EXCLUDE TABLE DATASET LIST

Reason: This is the header message for the list of EXCLUDE TABLE datasets displayed as a result of the DISPLAY,EXCLUDETB command.

Action: None

IAML0067 DSN: dsname or NO MATCHING DATASETS FOUND

Reason: This message is written once for every dataset in the Exclude Table that matches the selection criteria or once if no datasets are found.

Action: None

IAML0068 NO SELECT TABLE DATASET LIST PRESENT

Reason: A DISPLAY,SELECTTB command was entered and no Select Table exists.

Action: None

80.05 CONTINUED . . .

IAML0069 NO EXCLUDE TABLE DATASET LIST PRESENT

Reason: A DISPLAY,EXCLUDETB command was entered and no Exclude Table exists.

Action: None

IAML0070 RELEASELOCKS COMMAND INVALID

Reason: An invalid RELEASELOCKS command was entered.

Action: Check the syntax of the command and re-enter it correctly.

IAML0071 CLOSEFILE COMMAND INVALID

Reason: An invalid CLOSEFILE command was entered.

Action: Check the syntax of the command and re-enter it correctly.

IAML0072 GETMAIN FAILED ON CLOSEFILE COMMAND

Reason: The GETMAIN for an OCQ needed for the CLOSEFILE command failed, the close is not done.

Action: Check the region size of the IAMRLS address space.

IAML0073 CLOSE COMMAND FAILED, O/C TASK NOT ACTIVE.

Reason: The OPEN CLOSE TASK was not active and the close command could not be processed.

Action: The QUIESCE,FORCE command may have been entered, if so all files will have been closed. If not then call Innovation for further assistance in diagnosing this problem.

IAML0076 – IAML0085 IAMWRKLD MESSAGES

IAML0076 EXTRA I/O TASK ATTACHED

Reason: An extra I/O subtask was attached to handle an increase in workload.

Action: None

IAML0077 I/O TASK ATTACHED

Reason: This message is issued once for every I/O subtask attached during initialization, till the MINIOTASK is reached.

Action: None

IAML0086 – IAML0090 IAMLOGER

IAML0086 RLSLOGDD NOT PRESENT, LOGGING DISABLED

Reason: The RLSLOGDD DD statement was not present in the JCL used to start the IAMRLS address space. Logging is disabled.

Action: If you want logging active, stop IAMRLS and add the RLSLOGDD DD and restart IAMRLS.

80.05 CONTINUED . . .

IAML0091 – IAML0099 IAMBDSNT

IAML0091 IAMBDSNT PROCESSING COMPLETED

Reason: The processing of the SELECT and EXCLUDE lists is complete.

Action: None

IAML0100 – IAML0149 TRACE MESSAGES

IAML0100 TRACE DWA GETMAIN FAILURE – TRACE REQUEST FAILED

Reason: The trace processor was unable to obtain storage for it's work area.

Action: Check the region size of the IAMRLS address space and increase if needed.

IAML0101 TRACE SUBTASK NOT ACTIVE – TRACE REQUEST FAILED

Reason: The TRACE subtask is not active, tracing cannot be activated.

Action: Check the RLSLOGDD and JESLOG to determine why the TRACE subtask was not started. Call Innovation for help in diagnosing this problem.

IAML0102 TRACE REQUEST COMPLETED SUCCESSFULLY

Reason: A TRACE request has completed successfully.

Action: None

IAML0103 TRACE REQUEST FAILED

Reason: A TRACE request has failed.

Action: Call Innovation for help in diagnosing the problem.

IAML0104 INVALID TRACE KEYWORD – TRACE REQUEST FAILED

Reason: An invalid keyword was entered on a TRACE request.

Action: Check the syntax of the command entered and re-enter it correctly.

IAML0105 TRACE ID REQUIRED. ENTER ID= OR CANCEL

Reason: A trace ID is required on a TRACE request.

Action: Enter a trace ID or CANCEL in response to the WTOR.

IAML0106 TRACE JOB AND/OR DSN REQUIRED. ENTER JOB=, DSN= OR CANCEL

Reason: A jobname and/or a dataset name is required for a TRACE request.

Action: Enter a JOB=jobname or DSN=dsname or CANCEL in response to the WTOR.

80.05 CONTINUED . . .

IAML0108 TRACE TYPE REQUIRED. ENTERE TR= OR CANCEL

Reason: The type of TRACE records to record is required on a TRACE request.

Action: Enter one or more of the record types in response to the WTOR, for example: TR=(IOS,IOE,BFR,EXCP,XTND,PC)

IAML0109 TRACE REQUEST MUST BE START, STOP, LIST OR CANCEL

Reason: One of the above keywords must be present in the TRACE request.

Action: Re-enter the TRACE request specifying either START, STOP, LIST, or CANCEL.

IAML0110 TRACE_ID=XXXXXXXX

Reason: Output from a TRACE request showing the TRACE ID.

Action: None

IAML0111 TYPE=IOS,IOE,BFR,EXCP,XTND,PC

Reason: Output from a TRACE request showing the TRACE TYPES.

Action: None

IAML0112 DSN=dsname

Reason: Output from a TRACE request showing the DSNAME being traced.

Action: None

IAML0113 JOBNAME=jobname

Reason: Output from a TRACE request showing the JOBNAME being traced.

Action: None

IAML0114 STEPNAME=stepname

Reason: Output from a TRACE request showing the STEPNAME.

Action: None

IAML0115 OUTDD=ddname

Reason: Output from a TRACE request showing the OUTPUT DD.

Action: None

80.05 CONTINUED . . .

IAML0116 TRACE_ID ALREADY ACTIVE. TRACE REQUEST FAILED

Reason: A TRACE request with the same TRACE ID was already active, the current request is terminated.

Action: If this is a new request use a different TRACE ID, if you want to change the old request, STOP it first then restart it with the new options.

IAML0117 SPECIFIED TRACE_ID NOT FOUND. TRACE NOT DEACTIVATED

Reason: A TRACE,STOP command was issued but an active TRACE with the ID requested was not active.

Action: None

IAML0118 NO ACTIVE TRACE REQUESTS FOUND

Reason: No active trace requests were found to list on the TRACE,LIST command.

Action: None

IAML0119 TRE GETMAIN FAILURE. TRACE REQUEST FAILED

Reason: A GETMAIN for a trace element failed causing the trace request to be failed.

Action: Check the region size of the IAMRLS address space, contact Innovation for assistance in diagnosing this problem.

IAML0120 TRACE OUTPUT OPEN FAILURE. TRACE REQUEST FAILED

Reason: The TRACE processor could not open the requested trace output file.

Action: Make sure the file is allocated and has sufficient space.

IAML0121 TRACE OUTPUT DCB GETMAIN FAILURE. TRACE REQUEST FAILED

Reason: The GETMAIN for the trace output DCB failed. The TRACE request is failed.

Action: Check the region size of the IAMRLS address space, contact Innovation for assistance in diagnosing this problem.

IAML0122 OUTPUT DD FOR TRACE_ID xxxxxxxx IS ddname

Reason: Output from a TRACE request showing the OUTPUT DD.

Action: None

IAML0123 IAM ADDRESS SPACE TRACE REQUEST:

Reason: Output header line for a trace request.

Action: None

80.05 CONTINUED . . .

IAML0124 TRACE_ID xxxxxxxx DEACTIVATED

Reason: The trace id in the message has been deactivated.

Action: None

IAML0125 TRACE_ID xxxxxxxx QUEUED

Reason: A trace request has been queued and will become active when the requested selection criteria have been met.

Action: None

IAML0126 TRACE_ID xxxxxxxx ALREADY ACTIVE

Reason: An active trace request is already active with the trace id requested on a new trace request.

Action: If this is indeed a new request, choose a new trace id.

IAML0127 INVALID TRACE KEYWORD IN PARMLIB – TRACE REQUEST FAILED

Reason: An invalid trace keyword was detected, the trace request fails.

Action: Correct the trace keyword in error.

IAML0128 INVALID DSN SPECIFIED – TRACE REQUEST FAILED

Reason: An invalid dataset name was specified on a trace request, the trace request is failed.

Action: Re-enter the trace request with a valid dataset name.

IAML0129 DATASET ALLOCATION FAILED

Reason: Allocation of the output dataset for a trace request failed.

Action: Make sure the output dataset exists and is available for use.

IAML0130 LOG OF USER TRACE REQUEST

Reason: Log entry of the trace request as entered by the user.

Action: None

IAML0150 – IAML0154 OPEN/CLOSE MESSAGES

IAML0150 dsname OPENED DD=dsname J=jobname S=stepname I=jobid

Reason: DSNAME X.Y.Z DDNAME DDDDDDDD under JOBNAME JJJJJJ and STEPNAME SSSSSSS using PROGRAM PPPPPPPP was successfully opened by IAMRLS.

Action: None

80.05 CONTINUED . . .

IAML0151 dsname OPEN FAILED DD=ddname J=jobname S=stepname I=jobid

Reason: The indicated dataset was not successfully opened by IAMRLS.

Action: None

IAML0152 dsname DYNALOC FAILED RC=nnnn DD=ddname J=jobname S=stepname I=jobid

Reason: The indicated dataset could not be dynamically allocated by IAMRLS.

Action: None

IAML0153 dsname CLOSED DD=ddname J=jobname S=stepname I=jobid

Reason: The indicated dataset was successfully closed by IAMRLS.

Action: None

IAML0154 dsname CLOSE FAILED DD=ddname J=jobname S=stepname I=jobid

Reason: DSNAME X.Y.Z DDNAME DDDDDDDD under JOBNAME JJJJJJ and STEPNAME SSSSSS using PROGRAM PPPPPPPP was not successfully closed by IAMRLS.

Action: None

IAML0200 – IAML0229 IAMBRLOK (LOCK MANAGER) MESSAGES

IAML0200 INSUFFICIENT STORAGE FOR ADDITIONAL RECORD LOCKS

Reason: IAMBRLOK was unable to obtain more storage for lock tables.

Action: Make sure IAMRLS has sufficient region size in the startup jcl.

IAML0201 INSUFFICIENT STORAGE FOR ADDITIONAL OWNER ELEMENTS

Reason: IAMBRLOK was unable to obtain more storage for owner elements.

Action: Make sure IAMRLS has sufficient region size in the startup jcl.

IAML0202 INSUFFICIENT STORAGE FOR DEADLOCK DETECTION

Reason: IAMBRLOK was unable to obtain more storage for deadlock detection tables.

Action: Make sure IAMRLS has sufficient region size in the startup jcl.

IAML0203 RECORD LEVEL LOCKING ALREADY INITIALIZED

Reason: A call has been made to IAMBRLOK to perform initialization and IAMBRLOK has detected that initialization has already been done.

Action: Call Innovation for help in diagnosing this problem.

80.05 CONTINUED . . .

IAML0204 INSUFFICIENT STORAGE FOR LOCK HASH TABLE

Reason: IAMBRLOK was unable to obtain more storage for a lock hash table.

Action: Make sure IAMRLS has sufficient region size in the startup jcl.

IAML0205 INSUFFICIENT STORAGE FOR RECORD LOCKS

Reason: IAMBRLOK was unable to obtain more storage for record lock tables.

Action: Make sure IAMRLS has sufficient region size in the startup jcl.

IAML0206 INSUFFICIENT STORAGE FOR LOCK OWNER TABLE

Reason: IAMBRLOK was unable to obtain more storage for lock owner tables.

Action: Make sure IAMRLS has sufficient region size in the startup jcl.

IAML0207 INSUFFICIENT STORAGE FOR LOCK OWNER POOL

Reason: IAMBRLOK was unable to obtain more storage for a lock owner pool.

Action: Make sure IAMRLS has sufficient region size in the startup jcl.

IAML0208 . INSUFFICIENT STORAGE FOR DEADLOCK DETECTION

Reason: IAMBRLOK was unable to obtain more storage for deadlock detection tables.

Action: Make sure IAMRLS has sufficient region size in the startup jcl.

IAML0209 INSUFFICIENT STORAGE FOR LOCK WORK AREA

Reason: IAMBRLOK was unable to obtain more storage for a lock work area.

Action: Make sure IAMRLS has sufficient region size in the startup jcl.

IAML0210 DEADLOCK CONDITION DETECTED, I/O REQUEST FAILED FOR J=jobname S=stepname DD=ddname

Reason: A deadlock condition was detected for DD=ddname,J=JOBNAME,S=STEPNAME. The I/O request is failed.

Action: Wait for the other user of the resource to end and resubmit the job.

IAML0211 LOCKS HAVE BEEN RETAINED FOR JOB ASID=xxxx J=jobname S=stepname I=jobid

Reason: In response to a DISPLAY,RETAINEDLOCKS, this message indicates for what job steps record locks were retained due to an abend.

Action: None, informational message only.

80.05 CONTINUED . . .

IAML0212 RL DSN=dsname ASID=xxxx J=jobid S=stepname I=jobid

Reason: In response to a DISPLAY,RETAINEDLOCKS,DSN. This message indicates the dataset name, and associated jobstep for which record locks have been retained. A group of these messages is preceded by message IAML0211.

Action: None, informational message only.

IAML0213 NO LOCKS RETAINED

Reason: In response to a DISPLAY,RETAINEDLOCKS command, this message indicates that there are no retained locks.

Action: None, informational message only.

IAML0214 RETAINED LOCKS FOUND, SEE RLSLOGDD

Reason: In response to a DISPLAY,RETAINEDLOCKS command. The list of retained locks is displayed in the RLSLOGDD.

Action: None, informational message only.

IAML0215 Dsmame LOCKED ASID=xxxx J=jobname S=stepname I=jobid

Reason: In response to a DISPLAY,CONTENTION command, indicates that the specified job has record locks held for the identified dataset.

Action: None, informational message only.

IAML0225 IAMBRLOK ENTERED FOR JOB STEP TERMINATION

Reason: IAMBRLOK was entered to perform cleanup of locks for a job step termination.

Action: None, informational message only.

IAML0226 IAMBRLOK JOB STEP END RELEASING LOCKS

Reason: IAMBRLOK has released record locks during a job step end.

Action: None, informational message only.

IAML0227 IAMBRLOK JOB ABENDED, RETAINING LOCKS

Reason: IAMBRLOK has retained record locks after a job abend.

Action: Insure that a backout recovery is performed for this file, then release the locks if not already done so by IAMRLS.

IAML0228 IAMBRLOK BATCH SYNCPOINT RELEASING LOCKS

Reason: IAMBRLOK has been called from the batch syncpoint exit to release locks following a syncpoint exit call from the batch program.

Action: None, informational message only

80.05 CONTINUED . . .

IAML0229 IAMBRLOK ENTERED FOR BATCH SYNCPOINT

Reason: IAM RLS lock manager has been called to release locks from a job issuing batch syncpoints.

Action: None, informational message only.

IAML0230 – IAML0249 IAMBTERM (JOB STEP TERMINATION RTN) (COMMON SUFFIX) ASID=XXXX J=JOBNAME S=STEPNAME I=JOBID

IAML0230 JOBSTEP NORMAL TERMINATION ASID=xxxx J=jobname S=stepname I=jobid

Reason: The indicated job step, which had IAM files opened to IAM RLS, has completed without an abend.

Action: None, informational message only.

IAML0230 JOBSTEP ABEND TERMINATION ASID=xxxx J=jobname S=stepname I=jobid

Reason: The indicated job step has abnormally terminated. If the job was accessing recoverable datasets, then any record locks that were held will be retained until a recovery is performed.

Action: None, informational message only.

IAML0230 ADDRSPACE ABEND TERMINATIONASID=xxxx J=jobname S=stepname I=jobid

Reason: IAMBTERM has detected an address space termination of a job that was processing IAM datasets under IAM RLS. This message will only appear if an address space fails without having gone through the step termination.

Action: None, informational message only.

IAML0232 RECORD LOCKS RETAINED ASID=xxxx J=jobname S=stepname I=jobid

Reason: Record locks held by the specified job were retained due to an abend or because of address space failure.

Action: Make sure that a file recovery is performed, which will release the retained locks.

IAML0233 QUEUED DYNAMIC BACKOUT REQUEST ASID=xxxx J=jobname S=stepname I=jobid

Reason: IAM RLS posted the dynamic job backout task to perform a backout on the indicated job step due to an abend or address space failure.

Action: Verify from subsequent messages that the backout was performed and that it released the retained locks.

IAML0234 TERMINATION PROCESSING COMPLETED ASID=xxxx J=jobname S=stepname I=jobid

Reason: IAM RLS step termination processing has completed for the indicated job step. This means that any record locks held by the job step were either released, or retained as needed.

Action: None, this is an informational message only.

80.05 CONTINUED . . .

IAML0240 RELEASE RETAINED LOCKS REQUESTED ASID=xxxx J=jobname S=stepname I=jobid

Reason: A RELEASELOCKS command has been issued for the indicated job. Any information not provided on the command will be printed as astericks.

Action: None, informational message only.

IAML0241 FOUND FOLLOWING MATCHING JOB ASID=xxxx J=jobname S=stepname I=jobid

Reason: In response to a RELEASELOCKS command, IAM RLS has found the indicated job with record locks.

Action: None, informational message only.

IAML0242 RETAINED LOCKS RELEASED FOR ASID=xxxx J=jobname S=stepname I=jobid

Reason: IAM RLS released the retained record locks held for the indicated job step.

Action: None, informational message only.

IAML0243 REQUESTED JOB NOT FOUND ASID=xxxx J=jobname S=stepname I=jobid

Reason: The job specified on a RELEASELOCKS command could not be found.

Action: Issue either a DISPLAY,RETAINEDLOCKS or DISPLAY,CONTENTION to get a list of job(s) which currently have locks, and supply more job identification on the RELEASELOCKS command.

IAML0244 NONE RELEASED, MORE THAN 1 MATCH ASID=xxxx J=jobname S=stepname I=jobid

Reason: In response to a RELEASELOCKS command, IAM RLS found more than one matching job. No record locks were released

Action: Reissue the RELEASELOCKS command with more specific job identification, as can be found on the preceding IAML0241 messages.

IAML0500 – IAML0550 JOURNAL PROCESSING MESSAGES

IAML0502 GETMAIN1 FAILED, JOURNAL PROCESSING TERMINATED

Reason: A getmain for the journal work area failed during journal processing initialization, journal processing is terminated.

Action: If journal processing is desired, make sure the IAMRLS address space has a big enough region size to satisfy it's storage needs and restart the address space.

IAML0504 BUFFER GETMAIN FAILED, JOURNAL PROCESSING TERMINATED

Reason: A getmain for the journal output buffers failed during journal processing initialization, journal processing is terminated.

Action: If journal processing is desired, make sure the IAMRLS address space has a big enough region size to satisy it's storage needs and restart the address space.

80.05 CONTINUED . . .

IAML0506 SYNAD MESSAGE

Reason: The synad exit was entered during I/O to a journal file.

Action: Call Innovation for help in diagnosing this problem.

IAML0507 LOCATE FAILED FOR JOURNAL DSN dsname

Reason: A locate failed for the specified journal dataset during the initialization process for journaling.

Action: Make sure the dataset names provided in the startup parameters are correct and that the datasets are properly cataloged.

IAML0508 OBTAIN FAILED FOR JOURNAL DSN dsname

Reason: An obtain failed for the specified journal dataset during the initialization process for journalling.

Action: Make sure the dataset names provided in the startup parameters are correct and that the dataset(s) exist on the volumes to which they are cataloged.

IAML0511 IAMBJRNL USING DSN – dsname

Reason: The journal processor has selected the specified dataset for journal use.

Action: None, informational message only.

IAML0512 JOURNALLING HAS BEEN DISABLED

Reason: Journaling has been disabled for IAMRLS processing either due to an initialization error or because no journal datasets have been provided and SYSLOGGER has not been selected.

Action: If journaling is desired, fix the initialization error or specify journal datasets or SYSLOGGER in the IAMRLS startup parameters.

IAML0513 CONNECT FAILED TO LOGSTREAM: logstreamid

Reason: IAMBJRNL was unable to connect to the specified journal logstream.

Action: Make sure the system coupling facility policies have been defined correctly and the SYSTEM LOGGER is available on your CPU.

IAML0514 INIT OF JOURNAL FILES FAILED

Reason: IAMBJRNL was unable to complete initialization of the journal files.

Action: Make sure the dataset names provided in the IAMRLS parmlib member for journal datasets is correct, and that they exist on dasd and have available space.

IAML0515 IAMBJRNL THE NEXT JOURNAL FILE IS NOT EMPTY, JOURNAL FILE SWITCH HAS FAILED.

Reason: A JSWITCH command was entered to force a journal switch or a journal dataset has filled causing an automatic switch, but the next journal dataset is not empty.

Action: This will be followed by message IAML0534 requesting you ready the next journal dataset and respond "U" or reply "Q" to terminate the IAMRLS address space.

80.05 CONTINUED . . .

IAML0516 IAMBJRNL DSN – dsname DEALLOCATED

Reason: A journal dataset was deallocated.

Action: The data in the indicated journal should be copied to another dataset on tape or dasd, and the indicated journal dataset should then be emptied.

IAML0517 IAMBJRNL DEALLOCATION FAILED FOR DSN– dsname

Reason: The deallocation of a journal dataset failed.

Action: Call Innovation for help in diagnosing this problem.

IAML0518 IAMBJRNL DSN – dsname ALLOCATED

Reason: A Journal dataset has been allocated for use.

Action: None, informational message only.

IAML0519 IAMBJRNL ALLOCATION FAILED FOR DSN– dsname

Reason: The allocation of a journal dataset has failed.

Action: IAMRLS should switch to the next journal dataset if possible. Call Innovation for help in diagnosing this problem.

IAML0520 IAMBJRNL CONNECTED TO LOGSTREAM: logstream

Reason: The name of the System Logger Logstream that IAMRLS connected to for journal processing is displayed.

Action: None, informational message only.

IAML0525 IAMDD ddname: SHUTDOWN IN PROGRESS OR JOURNAL PROCESSING TERMINATED

Reason: The journal exit tried to journal a request for IAMDD ddname and was unsuccessful due to IAMRLS shutting down or the IAMRLS journal processing was terminated.

Action: Check the RLSLOGDD and JESLOG for other messages indicating why journal processing has been terminated or why IAMRLS is shutting down. Call Innovation for help in diagnosing this problem.

IAML0526 IAMDD ddname: GETMAIN1 FAILED, JOURNAL PROCESSING TERMINATED

Reason: The journal exit, IAMJRNRL, could not obtain storage for a journal work area for IAMDD ddname.

Action: Journaling for this dd is terminated. Make sure that the IAMRLS address space has enough region size specified at startup time.

IAML0527 IAMDD ddname: SAVEAREA GETMAIN FAILED, JOURNAL PROCESSING TERMINATED

Reason: The journal exit, IAMJRNRL, could not obtain storage for a save area to schedule a journal request to the journal subtask.

Action: Journaling for this dd is terminated. Make sure that the IAMRLS address space has enough region size specified at startup time.

80.05 CONTINUED . . .

IAML0528 IAMDD ddname: IXGWRITE FAILED, RC= nn ,RSNCD= nnnnn

Reason: The journal exit has received a failing return code from the System Logger during a write operation.

Action: Call Innovation for help in diagnosing this problem.

IAML0529 IAMDD ddname: JOURNAL UPDATE FAILED

Reason: The journal subtasks work area could not be found to schedule a journal request for IAMDD ddname.

Action: The journal request is not scheduled and the journal entry is lost. Contact Innovation for help in diagnosing this error.

IAML0530 IAMDD ddname: JOURNAL PROCESSING TERMINATED, JOURNAL ENTRIES LOST.

Reason: There is a problem with the journal buffer queue and a journal entry has not been able to be written.

Action: Check the RLSLOGDD and JESLOG for other messages indicating the source of the journaling problem. Call Innovation for help in diagnosing this error.

IAML0531 IAMDD ddname: IXGWRITES FAILING, JOURNAL PROCESSING TERMINATED

Reason: The journal exit is receiving failing return codes from the System Logger. Journal processing is terminated.

Action: Call Innovation for help in diagnosing this problem.

IAML0533 IAMBJRNL INITIALIZATION OF JOURNAL FILES FAILED, IAMRLS WILL QUIESCE.

Reason: The journal subtask was unable to initialize any of the journal datasets provided in the IAMRLS parmlib.

Action: Make sure the dataset names provided in the IAMRLS parmlib are correct, that the datasets exist on dasd and have available space.

IAML0534 READY DSN dsname, THEN REPLY 'U', OR REPLY 'Q' TO TERMINATE IAMRLS

Reason: The next journal dataset was not empty during a switch operation.

Action: Empty dataset dsname, then reply 'U' to continue journal processing, or reply 'Q' to terminate the IAMRLS address space.

IAML0551 – IAML0560 IAMSYNC MESSAGES

IAML0551 IAMSYNC STORAGE OBTAIN FAILED, NO SYNCPOINT RECORD WRITTEN

Reason: IAMSYNC was unable to obtain storage for a work area and will not be able to write the syncpoint record for the batch program that called it.

Action: Make sure the batch job has sufficient region size to satisfy the storage request.

80.05 CONTINUED . . .

IAML0552 IAMSYNC ATTEMPT TO LOAD IAMASPT FAILED, NO SYNCPOINT RECORD WRITTEN

Reason: IAMSYNC was unable to load module IAMASPT to perform syncpoint processing.

Action: Check the LINKLIST/JOBLIB/STEPLIB and make sure that the batch job has access to the IAM load library where module IAMASPT is located.

IAML0561 – IAML0570 IAMASPT MESSAGES

IAML0561 IAMASPT UNABLE TO LOAD IEANTRT, NO SYNCPOINT RECORD WRITTEN

Reason: IAMASPT was unable to load the IBM module IEANTRT to perform name token services. Without this module IAMASPT cannot write a syncpoint record.

Action: Make sure that IBM's common services library, SYS1.CSSLIB, is in the LINKLIST or that programs have access to it.

IAML0562 IAMASPT RECEIVED BAD RETURN CODE FROM TOKEN SERVICES, NO SYNCPOINT RECORD WRITTEN

Reason: IAMASPT was unable to retrieve the GLobal Token Area for the IAMRLS address space and cannot perform SYNCPOINT services.

Action: Check to see that IAMRLS is active, if not start it and restart the job. If it is active, call Innovation for help in diagnosing this problem.

IAML0563 IAMASPT RECEIVED BAD RETURN CODE FROM ALESERV EXTRACT, NO SYNCPOINT RECORD WRITTEN

Reason: The IAM batch syncpoint process encountered an error when invoking the IBM ALESERV service.

Action: Contact Innovation for assistance.

IAML0564 IAMASPT HAS TRIED TO EXCEED THE LIMIT OF 103 FILES, NO SYNCPOINT RECORD WRITTEN

Reason: The IAM batch syncpoint process has a limit of 103 IAM files.

Action: Either reduce the number of files that the program is using, or contact Innovation for assistance.

IAML0565 IAMASPT FOUND IAM RLS MARKED AS INACTIVE IN THE GLOBAL TOKEN Area, NO SYNCPOINT RECORD WRITTEN

Reason: IAMRLS has been marked as inactive and no syncpoint services are available.

Action: Determine why the IAMRLS address space is inactive and if possible, restart IAM RLS then restart the batch job.

IAML0566 IAMASPT FOUND IAM RLS MARKED AS QUIESCE FORCE IN THE GLOBAL TOKEN, NO SYNCPOINT RECORD WRITTEN

Reason: A QUIESCE,FORCE condition exists in the IAMRLS address space and it is longer accepting requests.

Action: Determine why the IAMRLS address space has been quiesced.

80.05 CONTINUED . . .

IAML0571 – IAML0580 IAMBSYNC MESSAGES

IAML0571 IAMBSYNC UNABLE TO OBTAIN STORAGE; NO SYNCPOINT WRITTEN

Reason: IAMBSYNC was unable to obtain storage for a work area, the syncpoint record could not be written.

Action: Make sure the job has a large enough region size when started to accommodate all of the storage required.

IAML0572 IAMBSYNC FOUND JOURNAL-IN-TROUBLE FLAG ON IN THE IWA, NO SYNCPOINT WRITTEN

Reason: The journal subtask has set a bit in the IWA indicating that initialization failed or that journalling is not active. The syncpoint record can not be written.

Action: Browse the RLSLOGDD and JESLOG for additional messages to determine why the journal subtask is not active.

IAML0573 IAMBSYNC FOUND AN ADDRESS OF ZERO IN THE GLOBAL TOKEN FOR THE JOURNAL'S TWA, NO SYNCPOINT WRITTEN

Reason: There was no task work area for the journal subtask in the Global Token Area. The subtask has either failed or was not activated.

Action: Browse the RLSLOGDD and JESLOG for additional messages to determine why the journal subtask has failed or is not active.

IAML0574 IAMBSYNC FOUND JOURNAL FULL BIT ON AFTER WAITING FOR THE JOURNAL WRITE, NO SYNCPOINT WRITTEN

Reason: There is a problem with the journal output buffers, the syncpoint record cannot be written.

Action: Call Innovation for help in diagnosing this problem.

IAML0575 IAMBSYNC IXGWRITE FAILED; RC=nn, RSNCD=nnnn, NO SYNCPOINT RECORD WRITTEN

Reason: IAMBSYNC attempted to write a syncpoint record to the System Logger, and the write failed with the indicated return and reason codes.

Action: Browse the RLSLOGDD and JESLOG for additional messages that might indicate why the System Logger Connection is not working. Call Innovation for help in diagnosing this problem.

IAML0576 IAMBSYNC'S CALL TO RELEASE THE LOCKS HELD BY THIS JOB FAILED

Reason: A call to release the record locks held by this job has failed.

Action: Call Innovation for help in diagnosing this problem.

80.05 CONTINUED . . .

IAML0581 – IAML8999 UNUSED
IAML9000 – IAML9100 IAMZAPR MESSAGES

IAML9000 IAMZAPR MAINTENANCE APPLIED

Reason: The ptf's supplied as input to the APPLY command have been successfully applied to the virtual storage copies of the IAMRLS processing modules.

Action: None

IAML9001 IAMZAPR MAINTENANCE RESTORED

Reason: The ptf's supplied as input to the RESTORE command have been successfully removed from the virtual copies of the IAMRLS processing modules.

Action: None

IAML9002 IAMZAPR MAINTENANCE NOT APPLIED

Reason: The ptf's supplied as input to the APPLY command have not been successfully applied to the virtual storage copies of the IAMRLS processing modules.

Action: Browse the output from SYSPRINT the DD and RLSLOGDD for additional information as to why the ptf's were not applied.

IAML9003 IAMZAPR MAINTENANCE NOT RESTORED

Reason: The ptf's supplied as input to the RESTORE command have not been successfully removed from the virtual storage copies of the IAMRLS processing modules.

Action: Browse the output SYSPRINT DD and RLSLOGDD for additional information as to why the ptf's were not removed

IAML9004 IAMZAPR ALLOCATION FAILED FOR DSN – dsname

Reason: The allocation of the dataset specified on either the apply or restore command has failed.

Action: Check the syntax of the command entered and make sure the dataset name is correct and is available. Re-enter the command with the correct dataset name.

IAML9005 IAMZAPR OPEN FAILED FOR DSN – dsname

Reason: The open of the dataset specified on either the apply or restore command has failed.

Action: Check the syntax of the command entered and make sure the dataset name is correct and is available. Re-enter the command with the correct dataset name.

IAML9006 IAMZAPR OPEN OF SYSPRINT DD FAILED

Reason: IAMZAPR was unable to open the SYSPRINT DD during APPLY or RESTORE command processing.

Action: Make sure the IAMRLS JCL used to start the IAM RLS address space has a SYSPRINT DD included. If it does, browse the RLSLOGDD, JESLOG, or SYSMSGS for additional messages documenting the reason for the open failure.

80.05 CONTINUED . . .

IAML9007 IAMZAPR OPEN OF SYSLIB DD FAILED

Reason: IAMZAPR was unable to open the SYSLIB DD during APPLY or RESTORE command processing.

Action: Make sure the IAMRLS JCL used to start the IAM RLS address space has a SYSLIB DD included. If it does, browse the RLSLOGDD, JESLOG, or SYSMSGS for additional messages documenting the reason for the open failure.

IAML9008 IAMZAPR VIFKEY MISMATCH BETWEEN IAMZAPR AND IAMVECTB FROM SYSLIB

Reason: There is a module level mismatch between the execution load library and the SYSLIB load library included in the startup JCL for IAMRLS.

Action: Make sure the SYSLIB DD points to the current IAM load library and is the same as the IAM load library in use from the LINKLIST/JOBLIB/STEPLIB.

IAML9009 IAMZAPR VECTOR TABLE FOR CURRENT LEVEL NOT FOUND, VIF NOT ACTIVE

Reason: IAMZAPR was unable to locate the IAM VECTOR TABLE for the current level of the VSAM INTERFACE that matches the level of the IAMZAPR module.

Action: Browse the RLSLOGDD and the SYSTEM LOG to determine why VIF was not started, or if it was stopped.

IAML9010 IAMZAPR PROCESSING PTF nnnnnnnnn

Reason: The PTF being processed by either the APPLY or RESTORE command is documented here.

Action: None, informational message only.

IAML9011 THE ABOVE STATEMENT IS MISSING AN OPERAND. REASON=9

Reason: IAMZAPR has encountered a missing operand in the PTF control statements.

Action: Correct the error with the PTF in the input library supplied to either the APPLY or RESTORE command and re-enter the command.

IAML9012 THE ABOVE STATEMENT CONTAINS AN ODD NUMBER OF DIGITS. REASON=15

Reason: The PTF input contains an odd number of digits in one of the offsets or data portions of the zap.

Action: Correct the error with the PTF in the input library supplied to either the APPLY or RESTORE command and re-enter the command.

IAML9013 THE ABOVE STATEMENT CONTAINS AN INVALID HEXADECIMAL DIGIT. REASON=16

Reason: The PTF input contains an invalid hexadecimal digit in one of the offsets or data portions of the zap.

Action: Correct the error with the PTF in the input library supplied to either the APPLY or RESTORE command and re-enter the command.

80.05 CONTINUED . . .

IAML9014 MODULE IAMVECTB NOT FOUND. REASON=19

Reason: The IAMVECTB module could not be found in LPA. The operation is failed.

Action: Browse the SYSLOG to see why VIF is not longer active.

IAML9015 THE ABOVE MODULE IS NOT ACTIVE. REASON=7

Reason: The module that was named on the PTF NAME statement was found in storage, but is marked as inactive.

Action: VIF must have been stopped, in order to apply the PTF's to the virtual storage copy of the module you will have to restart VIF.

IAML9016 THE ABOVE MODULE WAS NOT FOUND. REASON=8

Reason: The module that was named on the PTF NAME statement was either not in the table of modules that can be zapped, or could not be loaded or found in storage.

Action: Call Innovation for help in diagnosing this problem.

IAML9017 NAME CARD MISSING. REASON=13

Reason: The PTF supplied as input to the APPLY or RESTORE command is missing a NAME statement.

Action: Check your input against the PTF that was sent to you and if both are missing the NAME statement, call Innovation for a corrected PTF.

IAML9018 THE ABOVE STATEMENT OFFSET IS INVALID. REASON=11

Reason: The offset in the PTF provided is invalid.

Action: Check your input against the PTF that was sent to you and if both are the same, call Innovation for a corrected PTF.

IAML9019 THE MODULE WAS FOUND IN PLPA OR IN READ ONLY NUCLEUS WHICH IS NOT VALID. REASON=11

Reason: The module named on the NAME statement in the PTF supplied as input to either the APPLY or RESTORE command was found in PLPA or the read only nucleus.

Action: None of the IAM modules should be in PLPA or the read only nucleus. If you have put IAM modules in PLPA then you cannot use the online zap command to apply maintenance against them, you must use superzap and IPL with a CLPA.

IAML9020 THE ABOVE STATEMENT OFFSET IS NOT WITHIN THE MODULE. REASON=12

Reason: The offset in the PTF provided is invalid.

Action: Check your input against the PTF that was sent to you and if both are the same, call Innovation for a corrected PTF.

80.05 CONTINUED . . .

IAML9021 MODULE EP ADDRESS = xxxxxxxx

Reason: The entry point of the module about to be zapped is documented here.

Action: None

IAML9022 VERIFY FAILED DATA = xxxxxxxx

Reason: The PTF failed to verify with the data printed in this message.

Action: Check your input against the PTF that was sent to you and if both are the same, call Innovation for a corrected PTF.

IAML9023 OLD DATA TO REPLACE= xxxxxxxx

Reason: The data being replaced in the module by the PTF is documented here.

Action: None

IAML9024 BASE CARD IS NOT SUPPORTED FOR LPA. REASON=14

Reason: There is a BASE card included with the PTF, which is not allowed for the intercept modules located in LPA.

Action: This PTF must be applied using superzap, and a recycle of IAMRLS and VIF scheduled to pick up the maintenance.

IAML9025 SETSSI DOES NOT CONTAIN 8 HEXADECIMAL DIGITS. REASON=24

Reason: There is an invalid number of digits in a SETSSI keyword in the PTF.

Action: Check your input against the PTF that was sent to you and if both are the same, call Innovation for a corrected PTF.

IAML9026 CHECKSUM IS xxxxxxxx

Reason: The checksum data in the PTF is documented here.

Action: None, informational message only.

IAML9027 THE CURRENT CHECKSUM VALUE IS................xxxxxxxxxx

Reason: The actual checksum of the zap statements is documented here.

Action: None

IAML9029 IAMZAPR WILL APPLY ZAPS TO LPA NOW

Reason: The checksum and verify data is correct, the zap will be applied to the virtual storage copy of the module.

Action: None

80.05 CONTINUED . . .

IAML9031 CHECKSUM IS CORRECT

Reason: The checksum data supplied in the PTF is correct.

Action: None

IAML9032 CHECKSUM IS WRONG – PLEASE CORRECT THE ZAP. REASON=1

Reason: The checksum data provided in the PTF is incorrect.

Action: Check your input against the PTF that was sent to you and if both are the same, call Innovation for a corrected PTF.

IAML9033 IF YOU DO NOT FIND THE ERROR, PLEASE CALL INNOVATION TECH SUPPORT (KEEP THE OUTPUT LISTING).

Reason: This comes out after the IAML9033 message to remind you call Innovation if you cannot find the error.

Action: Call Innovation if you cannot find the error.

IAML9034 THE FOLLOWING PAGES CONTAIN THE PARTIAL CHECKSUM OF EVERY STATEMENT (FOR INNOVATION USE ONLY)

Reason: Documentation of the error in applying the PTF.

Action: Save for Innovation Tech Support.

IAML9035 THE ABOVE STATEMENT IS INVALID. REASON=10

Reason: The PTF statement printed above this message is incorrect.

Action: Check your input against the PTF that was sent to you and if both are the same, call Innovation for a corrected PTF.

IAML09100– IAML9999 UNUSED

80.10 IAM ABEND CODES

As a general rule, the IAM access method avoids intentionally abending, but rather passes return codes and error codes back to the calling program. Most of the abend codes listed below are for the various utility programs available with IAM, including IAMRECVR, IAMZAPOP, IAMSMFVS, and IAMSMF.

The abend codes issued by the old IAM Native and ISAM interfaces have been removed from the manual. They are documented in the ICL library, that was loaded as part of the product installation. Refer to member OLDABEND.

ABEND CODE	DESCRIPTION
U0102	The UPAD exit routine returned to IAM with the contents of R1 being either x'00', or not pointing to the parameter area that was passed from IAM to the UPAD exit.
U0184	An error condition occurred during IAM processing, and the user had supplied a //IAMDEBUG DD DUMMY DD statement. For various errors, IAM will check if an IAMDEBUG DD has been supplied, and rather than returning an error code and non-zero return code, IAM will issue an abend. The primary purpose of this capability is to add in problem diagnosis.
U0185	IAM-RLS was unable to obtain the requested record lock, and the RLS LOCK parameter indicated that it should abend the job. R15 has the associated logical error code. See Section 80.21 for the meaning of the code in R15.
U0402	An IAM utility program encountered a failure attempting to open the required SYSPRINT DD statement. Most likely the DD statement is either missing, or incorrectly spelled.
U0502	An IAM utility program encountered an error processing the required control card input. The required SYSIN DD statement is missing or incorrectly spelled, or no control cards were supplied. There should be an error message provided on the SYSPRINT listing indicating the cause of the problem.
U0600	An IAM utility program encountered an error processing an input or output file. Refer to SYSPRINT for messages detailing the error encountered.
U0658	An IAM utility program encountered a module that did not match the level of the utility program being executed. This may indicate an incorrect STEPLIB, or a problem with the product installation.
U0659	An IAM utility program encountered an unexpected logical error during processing. There should be messages on SYSPRINT further explaining the cause of the problem.
U0660	An IAM utility program encountered an internal save area stack overflow. Contact Innovation for assistance.
U0777	IAMSTART encountered an error with the SETLOCK service. Contact Innovation for assistance.
U0900	The IAM utility IAMEXTND, encountered a correctable user error. Refer to the SYSPRINT output for messages detailing the cause of the error. Contact Innovation.
U0901	The IAM utility IAMEXTND encountered a serious error. Refer to the SYSPRINT listing for messages indicating the reason for the failure. Contact Innovation.
U0902	The IAM utility IAMEXTND encountered a serious processing error. Refer to the SYSPRING listing for error messages. Contact Innovation.

80.20 CATALOG RETURN CODES

The following is a list of return codes and reason codes that IAM will set for a file DEFINE. When a file is defined under IDCAMS, these codes appear in the IDC3009I message. IDCAMS may print out other messages that relate to the failure code. When ever possible, IAM uses codes that will have the same or similar meaning for VSAM files, however that is not always possible. There will be an IAM error message, with the IAMW prefix that should identify the error in more detail. Also, for most allocations, there will also be error messages generated by the failed dynamic allocation request, which IAM will print out. Due to the way IDCAMS displays messages, the error messages printed by IAM will actually appear before the card images for the actual DEFINE. Then, IDCAMS will print out it's own error messages based upon the codes that IAM set.

RETURN CODE	REASON CODE	ERROR DESCRIPTION
8	6	Invalid type specified on a DELETE request.
8	38	Duplicate data set name found in the catalog, or on the volume(s) to which the data set is being defined.
16	0	SMS has failed the allocation request. Refer to the associated SMS error messages for additional information on the exact cause of the error.
22	8	The user attempted to define a cluster of a type not supported by IAM. Presently, IAM only supports KSDS or ESDS type of clusters without any alternate index.
42	nnnn	MVS/ESA DADSM allocation of the data set failed. The reason code is the return code from DADSM.
44	12	The work area provided by the caller of a request for information from the catalog for an IAM file was not large enough to contain all of the requested information.
54	nnnn	MVS/XA DADSM allocation of the data set failed. The reason code is the return code from DADSM.
56	6	User is not authorized to define the data set, according to the security system.
58	nnnn	On a DEFINE RECATALOG request, the attempt to OBTAIN the VTOC information for the specified data set failed. The reason code is the return code from the OBTAIN service.
58	4	A CVAF service request issued by IAM during the define of an IAM file failed. CVAF indicated that the volume on which the data set was defined was not mounted. There should be a corresponding IAMW50 error message.
58	8	A CVAF service request issued by IAM during the define process of an IAM file failed. CVAF indicated that the DSCB for the IAM file was not found on the volume to which the IAM file was defined. There should be a corresponding IAMW50 error message.
58	12	A CVAF service request issued by IAM during the define process of an IAM file failed. The CVAF return code and reason code are on the associated IAMW50 error message.
60	4	A catalog information request was issued (locate SVC) which appeared to be for an IAM file, however either an error occurred during IAM processing, or the file is not an IAM file. Normally, this return code will only be set if an //IAMDEBUG DD DUMMY statement is included in the JCL.
62	0	The initialization of the IAM file being defined failed. There should be IAMWxx messages indicating the cause of the error.
62	84	IAM found an error within the block of the file containing the alternate index and path information.
62	88	There was insufficient space to add additional alternate index or path information. Using a larger block size for the base cluster may correct the problem.
68	20	The define request failed because there was insufficient space on the specified volume(s) to contain the data set.
72	4	During a define of an IAM file, IAM was not able to find one or more of the requested volumes online.
80	4	During a define of an alternate index or path, the related data set or path entry data set was not found. Make sure that the related data set name or path entry data set name is specified correctly.

80.20 CONTINUED . . .

RETURN CODE	REASON CODE	ERROR DESCRIPTION
80	10	A define of an alternate index or path was missing the related or path entry name.
80	12	During a define of an alternate index or path, the related data set or path entry specified was invalid for the type of data set being defined. Make sure that the related data set is an IAM data set of the appropriate type (base for an alternate index define, and base or alternate index for a path define.)
86	4	During a define recatalog of an IAM data set, IAM encountered a failure attempting to determine the attributes of the file. The file most likely is not an IAM file.
86	6	During a define recatalog of an IAM data set, IAM could not find the data set on the specified volume.
96	4	Maximum record size for dataset exceed IAM's maximum record size.
96	8	The base cluster record size is not large enough to contain the specified alternate key, or the calculated minimum alternate index record size exceeds the record size defined for the alternate index.
132	xx	During the define of an IAM file, the internal parameter list usually generated by IDCAMS was missing data. The reason code indicates the particular field that was not provided. This most likely is not a user error, but rather an error on the part of the software issuing the define.
132	2	No VOLUME FVT was found in the parameter list.
132	4	No AMDSB FVT was found in the parameter list.
132	8	No Average LRECL FVT was found in the parameter list.
132	10	No Space FVT was found in the parameter list.
132	26	No SPACE was found the FPL.
132	34	No attribute FPL was found (RGATTR FPL) on the define of an IAM Path or alternate index.
132	48	No AMDSB in the FPL.
136	2	No VOLUME information length.
136	6	No CLUSTER FVT found.
136	18	No Average LRECL found in the FPL.
140	36	The IAMOVRID override statements contained an error, either in syntax or an unknown keyword was specified.
140	72	Duplicate volume found in volume list.
168	2	IAMís internal device characteristics table does not contain an entry for the type of device on which the user is attempting to define an IAM file. If the device is a valid DASD device, contact Innovation for support.
176	0	The define of an IAM file failed because there was insufficient room in the VTOC on the specified volume(s) to contain an entry for the new file. The data set will have to be allocated on a different volume, or some data sets will need to be deleted from the target volume. Another option is to enlarge the size of the VTOC.
184	4	The define of an IAM file failed because the dataset was in use by another job.
192	0	During the file define, IAM determined that the maximum record length being requested is longer than IAM supports for files without the SPANNED attribute. IAM can support records up to 32,760 bytes long if they are not spanned.
240	4	IAM was unable to determine the device characteristics for the volume requested on the file define request.
240	36	On the file define, the required DD statement for the volume(s) on which the data set is to be defined was not found, or was not provided. This return code should normally not be seen by an end user.

80.21 IAM I/O REQUEST ERROR CODES

IAM adheres to the VSAM application programming interface guidelines. When a request fails, IAM sets a non-zero return code in register 15, and provides a reason code within the RPL. Abends are avoided as much as possible. The appropriate exit routine, EODAD, LERAD, or SYNAD will be given control if so specified by the application program. It is the responsibility of the application program to verify the results of each I/O request, and take the action it deems appropriate in response to any error circumstance. As a result of a failing request, IAM will set the return code and error code to match the VSAM codes as much as is possible.

The table below indicates the return code, which is returned in register 15, and the error code, which is returned in the RPL field RPLERRCD. (The return code is also in the RPL, in field RPLRTNCD.) The error code is returned to an application program through the use of the SHOWCB macro, by requesting the FDBK field of an RPL.

RETURN CODE	ERROR CODE (DECIMAL)	ERROR CODE (HEX)	DESCRIPTION OF ERROR CONDITION
0			Request completed successfully.
0	8	X'08'	Indicates a duplicate alternate key. For GET requests, this means that there are additional base records with the same alternate key. For PUT requests, indicates a duplicate entry for the alternate key was created.
4	0	X'00'	Indicates that the RPL was already active with another request.
08			Logical Error Occurred (See Error Code for Reason)
08	4	X'04'	Logical End of File, there are no records with any higher key value than that of the last record returned. For a POINT, or START BROWSE, the key specified is higher than the highest key on the file.
08	8	X'08'	Duplicate Record. A PUT to add a record was issued, however there was already a record on the file with that same key.
08	12	X'0C'	Out of sequence. On a PUT during file load, the key of the record is lower than the previous record. Records must be loaded in ascending key sequence. On a PUT in sequential mode (OPTCD=SEQ), the key of the record being added is lower than the key of the last record processed (either retrieved or added) by this RPL. On a Skip Sequential request (POINT or GET), the key requested is lower than the key of the record previously retrieved.
08	16	X'10'	No record was found in the file with the specified key.
08	20	X'14'	Record is under Exclusive Control: The same record has been requested for UPDATE by another RPL. The RPL message area, if provided, has the address of the RPL that holds the requested record for update. From IAM-RLS: The records is locked by another RPL for the same logical unit of work (batch job, TSO user, or CICS transaction).
080	21	X’15’	IAM-RLS detected that by waiting for this record lock, this request would have caused in a deadlock.
08	22	X’16’	IAM-RLS: Request timed out waiting for a record lock.
08	24	X'18'	'IAM-RLS: Request was waiting for a record lock that was retained by another job due to an abend.

80.21 CONTINUED . . .

RETURN CODE	ERROR CODE (DECIMAL)	ERROR CODE (HEX)	DESCRIPTION OF ERROR CONDITION
08	28	X'1C'	For Compatible format files, the Independent Overflow area is filled, the file needs to be reorganized. For Enhanced format files, there either is insufficient DASD space to expand the file, or the file has used up the maximum number of extents it is permitted.
08	32	X'20'	ESDS: RBA supplied does not specify the address of the beginning of a record. No record at the specified relative byte address.
08	40	X'28'	IAM was not able to obtain virtual storage to complete the request. This will normally only occur on a GET request, with OPTCD=LOC, where IAM could not obtain storage for an area to contain the requested record.
08	44	X'2C'	The area provided by the application program was not large enough to contain the requested record. The record size is in the RPL field RPLRLEN.
08	64	X'40'	For Enhanced Format files: IAM was unable to obtain virtual storage for an additional string (place holder). For Compatible Format files, an insufficient number was specified for STRNO, and IAM ran out of place holders.
08	68	X'44'	An UPDATE request was issued, i.e. PUT or ERASE, however the file was opened for INPUT processing only.
08	72	X'48'	Keyed access attempted on an ESDS type of file.
08	80	X'50'	Erase attempted on an ESDS type of file.
08	84	X'54'	PUT with locate mode (OPTCD=LOC) is not permitted.
08	88	X'58'	RPL is not positioned for the specified sequential request. A POINT is required, or a random get with positioning: OPTCD=(DIR,NSP).
08	92	X'5C'	A PUT or ERASE request was issued without a preceding GET for update.
08	96	X'60'	On an update PUT request, the key in the record does not match the key of the record read for update.
08	100	X'64'	ESDS file type: on an update PUT request, the user attempted to change the record length.
08	104	X'68'	Invalid RPL options specified. (OPTCD) KSDS: Relative Byte Address or Control Interval processing is not supported by IAM. (OPTCD=ADR or OPTCD=CNV) or a get previous request is issued in skip sequential mode (RPL OPTCD=(SKP,BWD).
08	108	X'6C'	The record length either is less than the minimum record length, which is (key length + key offset), or exceeds the maximum defined record length. For compatible format IAM files with the FIXED length record attribute, the record length was not equal to the defined length.
08	112	X'70'	The key length in the RPL is greater than the defined length of the key. (RPL OPTCD=GEN type requests only.)
08	116	X'74'	The request type (i.e., a GET, POINT, or ERASE) is not valid during a file load.
08	144	X'90'	An invalid pointer was found in an alternate index. While the requested key was found in the alternate index, the associated base record was not found.

80.21 CONTINUED . . .

RETURN CODE	ERROR CODE (DECIMAL)	ERROR CODE (HEX)	DESCRIPTION OF ERROR CONDITION
08	156	X'9C'	An internal IAM error was encountered while processing the request. IAM found a record with a length of zero, or reached the end of data within a block unexpectedly. This error could be due to a corrupted buffer in storage. Run an IAMRECVR DIAGNOSE to verify that the file is ok.
08	185	X’B9’	I/O request for IAM-RLS was rejected because either IAM-RLS was no longer active, or was being quiesced.
08	189	X’BD’	I/O request for IAM-RLS was rejected because the file had been closed under IAM-RLS by operator request.
08	200	X'C8'	A control interval request (OPTCD=CNV) is not permitted when reading an ESDS through an alternate index.
08	208	X'D0'	A CHECK or an ENDREQ request was issued, however there was another task waiting on the RPL ECB.
08	225	X'E1'	Internal IAM error: The buffer pointer in the PLH is 0.
08	241	X'F1'	A invalid type of request was made, the contents of R0 contains a VSAM request type that IAM does not recognize or support.
08	242	X'F2'	The ECB address passed in the RPL is invalid.
08	244	X'F4'	During a file load, the Data Space used to temporarily hold the index was filled. Rerun job with the DATASPACE override, providing a larger value.
0C			Physical I/O error occurred. Message IAMW37, or IAMW01 should be examined to determine the nature of the error.
0C	4	X'04'	An error occurred attempting to READ a data block from DASD.
0C	16	X'10'	An error occurred when IAM was attempting to WRITE a data block to DASD.

80.22 IAM OPEN AND CLOSE ERROR CODES

In keeping with the VSAM application programming interface, most errors that occur during Open or Close will not cause an abend. Rather, a non-zero return code is passed in register 15, and an error code is set in the ACBERFLG field of the ACB. The error code can also be retrieved by the SHOWCB macro, requested for the failing ACB. Request FIELDS=ERROR in the SHOWCB macro to obtain the error code. Most of the IAM Open or Close failures will result in an IAMWxx error message being generated. Refer to the message for additional information.

RETURN CODE	ERROR CODE (DECIMAL)	ERROR CODE (HEX)	DESCRIPTION OF ERROR CONDITION
04			The CLOSE for the IAM file failed. See reason codes below. This could also be returned on OPEN to indicate that there was a warning message issued for one or more files being opened. The ACBERFLG should be checked for each ACB being opened.
04	4	X'04'	A CLOSE was requested for an ACB that was already closed.
04	100	X'64'	This is a warning that on an open for a PATH or base cluster with an alternate index(es) in an upgrade set, one or more of the alternate index data sets were found to be empty.
04	136	X'88'	Insufficient storage in region to close the file. Most likely, there is insufficient below the line storage to obtain work area(s) for the file close. Perhaps by closing other files first, if possible, will correct this circumstance. It may be necessary to raise the value of the REGION parameter for this job step.
04	144	X'90'	The CLOSE for an IAM file being loaded or reorganized failed, due to insufficient DASD space to write out the file's index. Delete and redefine the file with more DASD space.
08			Open Error Occurred. The file is not opened, see reason codes below for further information.
08	128	X'80'	DD Statement for the specified data set is not in the JCL. This also will be issued when one of the macros issued by IAM to OPEN the data set fails, such as OPEN, DEVTYPE, or TRKCALC.
08	136	X'88'	Insufficient storage in region to open the file. Most likely, there is insufficient extended private storage, however there could also be insufficient below the line storage. Check message IEF374I to determine how much storage was used. Most likely increasing the REGION parameter will solve the problem.
08	152	X'98'	RACF, or other system security software, indicated that the user was not allowed the type of access to the file that was being requested by the ACB on the OPEN macro.
08	160	X'A0'	The application attempted to open an IAM file for input processing, however either the file had never been loaded, or an attempted file load or reorganization failed. Or, the application attempted to open an unloaded file with a STRNO value not equal to 1.
08	167	X'A7'	IAM-RLS has encountered an abend attempting to open this dataset.
08	168	X'A8'	The specified file was already being processed by another job or user, and could not be opened for the processing that the application requested. Check for an IAMW04 or IAMW30, which will, when possible, include the name of the job that had the file opened. Rerun the failing job after the identified job terminates.
08	182	X'B6'	The Open was failed because the IAM-RLS address space was not active.
08	184	X'B8'	An I/O error occurred while trying to open the indicated file. This error code should be accompanied by either an IAMW37 or an IAMW07 error message, indicating the nature of the I/O error.
08	188	X'BC'	Open processing encountered a problem while attempting to Open the file. If the file was being loaded, an invalid or unsupported record length was specified for the file. This error may also be due a problem with the extended data areas of the file. There should be an IAMWxx message indicating the reason for the error.

80.22 CONTINUED . . .

RETURN CODE	ERROR CODE (DECIMAL)	ERROR CODE (HEX)	DESCRIPTION OF ERROR CONDITION
08	192	X'C0'	Open processing failed for a file load, due to an invalid file attribute being specified. This should be accompanied by an IAMW20 error message, indicating the improperly specified attribute.
08	196	X'C4'	Open processing failed on a PATH because the alternate index object of that PATH was empty.
08	209	X'D1'	The open for this file failed under IAM-RLS. Review the IAM-RLS log for additional information.
08	232	X'E8'	An attempt was made to Open a loaded IAM file for reloading with ACB MACRF=RST, however the file was defined with the NOREUSE attribute, and the IAM Global Options Table was set to ENABLE=NOREUSE. The file must be deleted and redefined to be reloaded.
08	240	X'F0'	The file was already open by another ACB (or DD) in this address space, and the current attempt to OPEN the file failed because the time stamp for file load was different than the time stamp in the previously opened ACB. Close all open ACB for this data set in this address space, and then they can be reopened.
08	241	X'F1'	The IAM VSAM Interface was not able to find a proper version of the IAM processing module required to OPEN this file.
08	242	X'F2'	The file was being opened for OUTPUT with an expired trial version of IAM.
08	248	X'F8'	Internal IAM error, the Open request failed.
08	254	X'FE'	The IAM VSAM Interface either was unable to successfully LOAD the required IAM processing module, or a version was loaded that does not match the version of the IAM VSAM Interface (VIF) that is active.

80.23 COBOL FILE STATUS CODES WITH IAM

While the above error codes are available to COBOL programs, more frequently these programs use the COBOL File Status Codes for interpreting error situations. Below are the more frequently encountered File Status Codes which can occur while processing an IAM file.

FILE STATUS	REASON FOR FILE STATUS CODE
00	Successful completion of request.
02	Successful completion of request. There are additional base records with the same alternate key.
04	Successful completion of request, however the record size does not match the fixed length of the record as defined within the COBOL program.
21	On sequential WRITES, the key is lower than the previously written key. Or, on REWRITE of an existing record, the application program changed the key of the record.
22	A WRITE of a new record was attempted for a key that matches a record currently in the file.
23	The requested record (key) was not found in the file.
24	Additional DASD space was required for this request, but it could not be obtained. For Compatible format files, this indicates that the Independent Overflow area has been filled. For Enhanced format files, either there was insufficient DASD space to expand the file, or the file had reached the maximum extents allowed, which is 16 extents per DASD volume. Generally, this error requires that the file be reorganized, possibly requiring a DELETE and DEFINE to allocate more DASD space. For Enhanced Format files, if the cause is insufficient DASD space, if some existing data sets can be deleted from the DASD volume, it might be possible to subsequently retry the request.
34	The file is full. See description for file status code 24 above.
39	This file status generally implies a mismatch between the defined IAM file attributes and the record layout in the COBOL program. For example, possible causes are the key length or offset from the file definition do not match the COBOL record layout. Also, the defined maximum record length for the file is shorter than the maximum possible length for variable length records. COBOL requires that the defined record length be at least as long as the maximum theoretical record size, from the layout. Refer to the IAM PSEUDOLRECL feature for circumventing this restriction.
41	An OPEN was requested for a file that had already been opened.
42	A CLOSE was issued for a file that was already closed.
43	A file update request (i.e., REWRITE or DELETE) was issued without a prior READ for UPDATE.
44	An incorrect record length was specified on the WRITE. Either the length was shorter than the minimum, which is key length plus key offset, or the record was longer than the defined maximum record size for the file.
46	A READ failed because the application had not successfully established a position in the file, or a READ was attempted after the end of file had been reached.
47	A READ request was issued for a file that was not opened for input or I/O.
48	A WRITE request was issued for a file not opened for output or I/O (update).
49	A DELETE or REWRITE request was issued for a file that was not opened for I/O (update).
92	Logic error. The file attributes from the COBOL program, such as key length, relative key position, or record length do not match the IAM file being accessed. (Note that the defined record length for the file must be at least as large as the largest possible record length.)
93	Insufficient virtual storage to process the request. Most likely, the REGION parameter has to be increased for the job. Check the IAM output, IAMPRINT, from a Listcat of the file to determine how much storage will be required to open this file.
94	Not positioned for a sequential READ request.
95	Invalid or incomplete file information.
96	The DD statement is missing for this file.

THIS PAGE WAS INTENTIONALLY LEFT BLANK

90.01 IAM INSTALLATION OVERVIEW

OVERVIEW The installation of IAM is quickly accomplished by following the procedure and instructions described in this section. The guidelines and tips provided will assist you in installing IAM, either as a new user, or as an upgrade from a prior version or maintenance release. Even if you have installed IAM previously, you are strongly encouraged to review and follow the install procedure and instructions from the manual that matches the version of IAM you are installing, as the procedures are modified on occasion, and default values can change.

INSTALL PROCESS The installation of IAM is accomplished by the following procedure:

1. Load the IAM Installation Control Library (Sec. 90.02).
2. Load the IAM program load library (Sec. 90.03).
3. Review and set the IAM Global Options. (Sec. 90.04)
4. (Optional) Load the sample JCL library. (Sec. 90.05).
5. (Optional) Install the IAM ISPF panels and programs. (Sec. 90.06)
6. (Optional) Install the IAM CICS monitor transaction. (Sec 90.07)
7. (Optional) Set up IAM Single System Record Level Sharing (Sec 90.10)
8. Activate IAM's VSAM Interface in your system (Sec 90.20).
9. Develop testing plans and procedures for IAM (Sec. 90.30).
10. Put IAM into production (Sec 90.40).

Upon completion of the above steps, you are ready to use IAM files in place of VSAM clusters. Whenever you specify the $IAM parameter in a DEFINE statement for a KSDS or ESDS cluster, you will create an IAM file instead of a VSAM cluster. For customers with the optional Alternate Index support, defines of alternate index and path datasets will be automatically converted to IAM when the base cluster is an IAM dataset. Any program referencing the file will be automatically switched to IAM instead of using VSAM.

PRODUCT TAPE The product distribution tape is in a SL format with a volume serial number of IAM80T or IAM80P. The tape volume IAM80T contains a trial version of the library which is date protected and will expire on the date shown on the external tape label. The trial version has the IAM alternate index feature enabled. The tape volume IAM80P contains the production version of the product library. The production version tape will have the alternate index option only if it has been purchased. There are no date protected modules in the production library.

TRIAL CUSTOMER If you are a trial IAM customer, the trial copy will expire on the date specified on the installation tape, unless it is extended by INNOVATION. You will be supplied a production library when you purchase the product. Reinstall the IAM system from the production tape before your trial expiration date.

CURRENT CUSTOMER If you are a current customer of IAM, you can install and test this new version of IAM without affecting any of your existing jobs. Programs can continue to use prior versions until you choose to have them use the new version. Refer to Sections 90.20 and 90.30 for information on testing the new level of IAM. Refer to Section 90.40 for procedures to safely and easily put the new version of IAM into production, and provide a safe backout procedure in case there are problems.

90.02 INSTALLING THE IAM INSTALLATION CONTROL LIBRARY

The first step to install the IAM product is to download the Installation Control Library (ICL) from the tape. This library will contain the JCL necessary to complete the rest of the install, and will have any updates to the various jobs since the manual was printed. Look for any members in the library that begin with NEWS for important product update information. The ICL is contained on file two of the installation tape, and is in IEBCOPY input format.

After copying the ICL to DASD, you should review member '@INDEX' in this library to determine which members might be helpful to you. Be sure to check the NEWS members as well.

ICL ATTRIBUTES The Installation Control Library must be loaded to a partitioned dataset on disk. You may load it to an existing dataset (if it has sufficient space and proper DCB attributes) or allocate and load a new one. The following table shows the allocation parameters for the Installation Control Library:

DATASET	RECFM	LRECL	BLKSIZE	BLOCKS	PDS DIR BLOCKS
IDP.ICLIAM80	FB	80	3120	150	23

This dataset is allocated in blocks so that the system will calculate the correct number of tracks for your device type. If your installation chooses a different blocksize for this dataset, you should adjust the number of blocks accordingly.

HOW TO INSTALL The JCL example below allocates and loads the Installation Control Library. You must make the following changes to reflect your environment:

1. 'DSN=IDP.ICLIAM80' on the ICLOUT DD statement should be changed to the name you wish to use for the Installation Control Library.
2. 'VOL=SER=vvvvvv' on the ICLOUT DD statement must specify a disk volume where the Installation Control Library will be allocated.
3. 'UNIT=TAPE' on the ICLIN DD statement must specify a tape drive capable of reading the installation tape.
4. 'VOL=SER=IAM80T' on the ICLIN DD statement must be changed to 'VOL=SER=IAM80P' if you are loading from a production installation tape.

After modifying the JCL, submit the job to download the ICL library.

90.02 CONTINUED . . .

JCL FOR LOADING THE ICL

```
//ICLLOAD    EXEC  PGM=IEBCOPY,REGION=1024K
//SYSUT3      DD   UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT4      DD   UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSPRINT    DD   SYSOUT=*
//ICLOUT      DD   DSN=IDP.ICLIAM80,                         ←USER-CHANGE
//               VOL=SER=vvvvvv,                             ←USER-CHANGE
//               UNIT=SYSDA,DISP=(,CATLG),
//               DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB),
//               SPACE=(3120,(150,15,23),,,ROUND)
//ICLIN       DD   DSN=IAMICL,
//               UNIT=TAPE,                                   ←USER-CHANGE
//               DISP=OLD,LABEL=2,
//               VOL=SER=IAM80T       CHANGE T TO P IF PRODUCTION TAPE
//*
//SYSIN       DD   *
  COPY I=((ICLIN,R)),O=ICLOUT
/*
```

Figure 1: JCL to down load the Installation Control Library (ICL)

Member ‘@INDEX' is the table of contents for this library. Installation Control Library provides the user procedures for:

- Installing the program load library. (Member IAMLOAD)
- Installing the Sample JCL library. (Member JCIAM80)
- Activating the IAM system level VSAM Interface.
- Verify the installation of IAM.
- Using IAM's VSAM ANALYSIS programs.
- And many more.

90.03 IAM PROGRAM LOAD LIBRARY

The second step is to download the IAM load library from the product distribution tape. JCL to perform this step is provided in the member IAMLOAD of the ICL library that was downloaded in the prior step (see Section 90.02).

The IAM program library is distributed on tape in IEBCOPY unloaded format. It is required that an authorized library be used, as certain modules are linked with an authorization code of 1. Note that normal application programs using IAM files do not require authorization. An authorized library is required when IAM is used for the following functions:

- Activation and deactivation of the IAM VSAM Interface.
- Activate and execute the IAM RLS address space.
- Job steps that use the IDCAMS utility on IAM files. This includes functions such as defining IAM files, listing catalog information, deleting IAM files, and copying IAM files with REPRO.
- Job steps that run the IAMRECVR utility program.
- When being used by other products or programs that require APF authorization.

The IAM Load Library must be explicitly authorized. This is done by making the necessary updates to SYS1.PARMLIB member IEAAPFxx or PROGxx for OS/390 and z/OS.

The IAM program library must be loaded to a partitioned dataset on disk. You may load it to an existing dataset (if it has sufficient space and proper DCB attributes) or allocate and load a new one. The following table shows the allocation parameters for the product program library:

LOAD LIBRARY ATTRIBUTES

PRODUCT	RECFM	LRECL	BLKSIZE	BLOCKS	PDS DIR BLOCKS
IAM	U	n/a	6144	360	36

This dataset is allocated in blocks so that the system will calculate the correct number of tracks for your device type. If your installation uses a different block size for this dataset, you should adjust the number of blocks accordingly.

HOW TO INSTALL

The JCL listed below is supplied on the Installation Control Library (Section 90.02) with a member name of 'IAMLOAD'. Be sure to use the JCL provided on this tape not JCL from prior releases. The provided JCL will have the correct volume for the version of the product that is being installed. Make the following changes to the supplied JCL:

1. 'DSN=IDP.MODIAM80' on the LOADOUT DD statement should be changed to the name you wish to use for the Load Control Library. For customers that have a prior version of IAM installed, **DO NOT INSTALL THE NEW VERSION INTO THE CURRENT PRODUCTION LIBRARY!** Please install the new version into a separate, authorized library. If the library is in the Link List, make sure that it is after the production library.
2. 'VOL=SER=vvvvvv' on the LOADOUT DD statement must specify a disk volume where the Load Library will be allocated.
3. 'UNIT=TAPE' on the LOADIN DD statement must specify a tape drive capable of reading the installation tape.

90.03 CONTINUED . . .

EXAMPLE JCL TO DOWN-LOAD THE IAM LOAD LIBRARY

```
//IAMLOAD     JOB   (IAM),'IAM-LOAD LIBRARY INSTALL'
//******
//*           THIS JOB INSTALLS THE INNOVATION ACCESS METHOD(IAM)
//*           TAPE VOLUME SERIAL MUST BE IAM80P
//*           USER CHANGES:
//*             CHANGE ONLY WHERE INDICATED BY   |←USER CHANGE|
//*           DSN=  USER LOAD LIBRARY
//*           UNIT= TAPE DRIVE UNIT NAME
//*           SER=  OUTPUT VOLSER FOR LOAD LIBRARY
//******
//*
//COPY        EXEC  PGM=IEBCOPY,REGION=1024K
//SYSPRINT     DD   SYSOUT=*
//SYSUT3       DD   UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT4       DD   UNIT=SYSDA,SPACE=(CYL,(1,1))
//LOADOUT      DD   DSN=IDP.MODIAM80,                    ←USER CHANGE
//            VOL=SER=VVVVVV,                            ←USER CHANGE
//            SPACE=(6144,(360,120,36),,,ROUND),
//            DCB=(LRECL=0,BLKSIZE=6144,RECFM=U),
//            DISP=(,CATLG),UNIT=SYSDA
//LOADIN       DD   DSN=IAMLOAD,
//            UNIT=TAPE,                                 ←USER CHANGE
//            DISP=OLD,LABEL=1,
//            VOL=SER=IAM80T          ←will reflect proper volser
//SYSIN        DD   *
  COPY INDD=((LOADIN,R)),OUTDD=LOADOUT
/*
```

Figure 2: JCL to down load the IAM Product Load Library

HOW TO INSTALL

After making the required changes to the member 'IAMLOAD' , submit the job to load the IAM program library from your installation tape.

****** CAUTION CAUTION CAUTION ******

1. For current IAM customers, we recommend that you install this version in an authorized library separate from your current IAM library. Once you have tested this version, then copy the modules into your production library replacing the previous version, or replace the previous IAM library name with the new one in your system's LINKLIST.
2. If you have installed the modules into a LINKLIST library, the operator may have to issue the command MODIFY LLA,REFRESH to inform the system of the update.

90.04 SET IAM GLOBAL OPTIONS

OVERVIEW The third step is to review and make whatever changes are necessary to the IAM Global Options. Many of the various default values for IAM can be changed in the IAM Global Options Table. For example, defaults for buffers, Data Compression, SMF recording and many others can be changed using this capability. The defaults are changed by executing the IAMZAPOP utility program. The use of this program and all of the default values and options are fully described in Section 91 of the IAM User Manual. This section will discuss what is new and changed in this version, how to use the IAM Global Options change capability along with some examples, plus make some recommendations for changing some of the defaults.

NEW IN IAM V8.0 New Global Options for Version 8.0 of IAM include:

- **RLS=([SHARE1 | SHARE2 | SHARE3 | SHARE4],[AND | OR],[TABLE])**
  This Global Option will set the criteria that will be used for automatically selecting datasets to be processed by IAM RLS. The default value is RLS=SHARE3, meaning IAM files with share option values of 3 or 4 will be automatically selected for IAM RLS processing, if IAM RLS is active.

NEW IN IAM V7.0 As a review for customers upgrading from Version 6.4 or earlier, the Global Options that were new or changed in Version 7.0 of IAM include:

- **CICSBUFSP=nnnnnn**–When running under CICS, IAM will use this value for Enhanced mode files to calculate a default MAXBUFNO. This will allow customers to set one value for storage usage for files opened under CICS, and a different value for batch. The reasoning for this is because many customers have much tighter storage limitations under CICS, than under batch jobs, so that batch can default to a larger value than what would be a desirable default for CICS. The default value is 256K (262,144 bytes), which is the same as the old default value for BUFSP.
- **REORGWTO**–This option, which can be enabled or disabled as desired, will control whether or not IAM will display the IAMW22 file reorganization recommended on the JOB log, when such a condition arises. The default value is that the messages will be enabled.
- **The BUFSP value has been increased to 875K (896,000).** This will allow IAM to set MAXBUFNO to a value large enough to hold slightly more than one cylinder of data. The value had previously defaulted to 256K. CICS will use the new Global Option CICSBUFSP for this calculation.
- **The CRBUFOPT default has been changed to MCYL.** This will automatically provide the best buffering possible for file loads. The value previously had defaulted to CYL.
- **The DATASPACE value has been increased to 256 megabytes.** This change was done to allow for larger Index Space storage to be obtained. The previous default was 128 megabytes.
- **The LOADABWO option is now defaulting to YES.** A value of YES is required for customers running CICS Transaction Server.
- **The MAXREGION default value has been increased to 512 megabytes.** The value had previously been set to 128 megabytes. The increase was made because many customers already had CICS regions in excess of 128 megabytes, which eliminated the benefit of this feature.
- **The VAROVERFLOW default has now been set to YES.** This value had defaulted to NO in Version 6.4 to allow for downward compatibility with Version 6.3.

90.04 CONTINUED . . .

USERS OF PRIOR VERSIONS OF IAM

The first step in determining what, if any, Global Options should be changed is to find out what Global Options were changed previously. User's of prior versions can determine what Global Options they have changed in their current release through the use of the IAMZAPOP AUDIT function. This is a critical part of installing a new release or level of IAM to insure that existing applications will continue to run as they have with the prior levels. Below is an example of running the AUDIT function.

EXAMPLE OF IAMZAPOP AUDIT COMMAND

```
//AUDIT     EXEC  PGM=IAMZAPOP
//STEPLIB    DD   DISP=SHR,DSN=current.iam.loadlib
//SYSLIB     DD   DISP=SHR,DSN=current.iam.loadlib
//SYSPRINT   DD   SYSOUT=*
//SYSIN      DD   *
            AUDIT
/*
```

Figure 3: JCL to run AUDIT Function of IAMZAPOP (EX9004A)

AUDIT COMMAND OUTPUT

An example of the output from the AUDIT command follows. Note that it prints the name of the load library. The changed Global Option(s) keywords are displayed, followed by the current value, and the default value is displayed. From this example, we see that the following Global Options were changed:

- RECTYPE=201
- SMF=YES

These options are important to maintain across releases, as they provide for IAM SMF data to be collected, and enable customer's to obtain reports on IAM activity using the IAMSMF or IAMSMFVS programs, or their own programs.

```
IAM400  IAM GLOBAL OPTION(S) CHANGE -- IAMZAPOP VER 8.0/00P  -- INNOVATION DATA PROCESSING  DATE - 2002.178  PAGE - 001
IAM303   CARD IMAGE - * AUDIT                                                                         00060000*
IAM491   AUDIT OPTIONS    FUNCTION STARTED - 08.12.31

IAM541   THE FOLLOWING OPTIONS HAVE BEEN CHANGED IN -- IAMX.TESTLIB

            RECTYPE          SMF RECORD TYPE ------------------------------201          DEFAULT---------000
            SMF              SMF RECORDING ENABLED ----------------------YES            DEFAULT----------NO

IAM547   AUDIT REQUEST COMPLETE FOR --- IAMX.TESTLIB
IAM492   AUDIT OPTIONS    FUNCTION ENDED   - 08.12.31 - CONDITION CODE 000

IAM499   IAMZAPOP(8.0/00P ) PROCESSING COMPLETED
```

Figure 4: Example of Output from AUDIT Command

90.04 CONTINUED . . .

RECOMMENDED GLOBAL OPTION CHANGES

The next step, after determining what Global Options had been changed for the prior version, is to review the various default values and determine which, if any, would provide a benefit to your installation. Section 91 of the IAM User Manual describes the various options available. Innovation recommends that you consider changing the following Global Option values from their default settings.

- Set **ESDSINTEGRATED=5** or higher to allow for updates to ESDS files to remain in their original block. While VSAM (and IAM) do not allow a record length to change on update processing for ESDS files, the record lengths can change due to data compression. It is better for overall performance to minimize the use of overflow, as much as is possible which is accomplished by defaulting to providing some integrated overflow for updated ESDS files.
- Set **SMF=YES** and **RECTYPE=201** to enable the IAM SMF records. This will provide a way, through SMF reports, to monitor and track IAM dataset usage. With this option enabled, the IAMINFO reports will be available from the IAM SMF records. This may prevent rerunning jobs just to obtain an IAMINFO report, should one be necessary. Valid record types are from 128 to 255. Make sure to select a record type that is not being produced by other software packages.

IAM GLOBAL OPTIONS AND OTHER SOFTWARE PRODUCTS

There are also a few options that you may have to set for compatibility with other software products. In particular, the following options may have to be set as indicated:

- Set **DYNCAT=YES** if you are using the BMC MAINVIEW product.
- Set **ENABLE=BIM** if you have any of the VSAM products from B. I. Moyle.
- Set **ENABLE=VAM** if you have CA-ALLOCATE, formerly SAMS (VAM) from Sterling.
- Leave **DSORG=PS** set, particularly if you use DFSMSdss or DFSMShsm to backup your datasets. If IAM files have a DSORG of DA set, these products may not be able to restore IAM datasets, particularly if they are multi-volume.

OTHER GLOBAL OPTION CONSIDERATIONS

There are a few other Global Options that should also be considered, which may need to be set depending on your installation requirements. These include the following:

- Consider increasing the **BUFOPNO** default value from 4 if there are many files with large overflow areas. This will increase the buffers that IAM starts with at OPEN time, and can reduce I/O and elapsed time to open IAM files with very large overflow areas.
- While rather rare, there are some applications that require the setting of the Global Option **ENABLE=NOREUSE** for proper function. Such applications require an open error to occur when opening a non-reusable VSAM dataset to prevent overlaying a previously loaded file. By setting this Global Option, IAM will follow VSAM rules for reloading datasets without a Delete and Define as based on the Define parameters.
- If you already have CICS regions of 512 megabytes or higher, then consider raising the default value for **MAXREGION** to allow IAM to increase the region of those CICS systems if necessary. With a default value of 512 megabytes, IAM will not be able to increase the region for CICS systems that already have the large region value.
- If X37 abends have been a problem, consider revising the **MAXSECONDARY** default values. This facility acts as a multiplier that IAM will use when requesting another extent. Because IAM files are limited to 16 extents per volume, use of this facility will increase the amount of DASD space requested without having to change the IDCAMS DEFINE. This option is not used for IAM files that are in the DFSMS Extended Format.
- Consider how you prefer allocations to work for multivolume datasets. Depending on this option, IAM will request either the primary or secondary space when it appears that a volume switch will occur. The keyword is **MULTIVOLUME=[PRIMARY | SECONDARY]**. Because IAM does not have control over whether or not a volume switch occurs, this may not always work as desired, particularly when the primary space value is less than the secondary. This option is not used for IAM files that are in the DFSMS Extended Format.
- If you have a DFSMS STORCLAS that causes datasets to not be SMS managed, such as NONSMS, then set the IAM Global Option STORCLAS to that value.

90.04 CONTINUED . . .

SETTING IAM GLOBAL OPTIONS

Once you have decided on the Global Options you want to change, use the program IAMZAPOP to change them. Global Options can also be changed subsequent to the product installation as may be needed. The following JCL and control card example demonstrates how to set the recommended Global Option values. Refer to section 91 for complete information on setting and changing the IAM Global Options Table.

```
//SETGLOPT EXEC   PGM=IAMZAPOP
//STEPLIB   DD    DISP=SHR,DSN=new.iam.loadlib
//SYSLIB    DD    DISP=SHR,DSN=new.iam.loadlib
//SYSPRINT  DD    SYSOUT=*
//SYSIN     DD    *
  ZAP             ESDSINTEGRATED=5,SMF=YES,RECTYPE=201
  PRINT
/*
```

Figure 5: Example of setting Recommended Global Option Values (EX9004B)

The above example also includes a PRINT command. This will print out the Global Option settings, and can be done whenever a list is necessary. It is a good idea to review this listing, to make sure that the Global Options are set as you expect them to be. The PRINT command does not need to be preceded by the ZAP command, so a listing of your Global Options can be obtained whenever necessary. This listing of your Global Options can also be obtained through the IAM ISPF panels.

90.05 INSTALLING THE SAMPLE JCL LIBRARY

SAMPLE JCL LIBRARY

The fourth step is to download the sample JCL library. Many of the examples of using IAM datasets provided in this manual are available in a JCL library that can be downloaded from the IAM Product Tape. The examples provided should be useful to all the various personnel that work with the IAM product. Installation of this library is optional, but highly recommended.

For ease of use, the member names will be the section number of the manual in which the example is found, suffixed by an alphabetic character. For example, the example in the prior section on setting the Global Options is member EX9004B, as it is in section 90.04, and is the second example. The member @INDEXJ will contain a listing and brief description of the contents of this dataset.

The Sample JCL Library must be loaded to a partitioned dataset on disk. You may load it to an existing dataset (if it has sufficient space and proper DCB attributes) or allocate and load a new one. The following table shows the allocation parameters for the Sample JCL Library:

SAMPLE JCL LIBRARY ATTRIBUTES

DATASET	RECFM	LRECL	BLKSIZE	BLOCKS	PDS DIR BLOCKS
IDP.JCLIAM80	FB	80	3120	150	23

This dataset is allocated in blocks so that the system will calculate the correct number of tracks for your device type. If your installation chooses a different block size for this dataset, you should adjust the number of blocks accordingly.

HOW TO INSTALL

The JCL below will allocate and load the sample JCL Library. This JCL is also in the ICL library that was down loaded from the tape, as member JCLIAM80. You must make the following changes to reflect your environment:

1. 'DSN=IDP.JCLIAM80' on the JCLOUT DD statement should be changed to the name you wish to use for the Installation Control Library.
2. 'VOL=SER=vvvvvv' on the JCLOUT DD statement must specify a disk volume where the Installation Control Library will be allocated.
3. 'UNIT=TAPE' on the JCLIN DD statement must specify a tape drive capable of reading the installation tape.
4. 'VOL=SER=IAM80T' on the JCLIN DD statement must be changed to 'VOL=SER=IAM80P' if you are loading from a production installation tape.

JCL TO INSTALL SAMPLE JCL LIBRARY

```
//JCLIAM80   EXEC   PGM=IEBCOPY,REGION=1024K
//SYSUT3      DD    UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSUT4      DD    UNIT=SYSDA,SPACE=(CYL,(1,1))
//SYSPRINT    DD    SYSOUT=*
//JCLOUT      DD    DSN=IDP.JCLIAM80,              ←USER-CHANGE
//           VOL=SER=vvvvvv,                       ←USER-CHANGE
//           UNIT=SYSDA,DISP=(,CATLG),
//           DCB=(LRECL=80,BLKSIZE=3120,RECFM=FB),
//           SPACE=(3120,(150,15,23),,,ROUND)
//JCLIN       DD    DSN=IAMJCL,
//           UNIT=TAPE,                            ←USER-CHANGE
//           DISP=OLD,LABEL=3,
//           VOL=SER=IAM80T    ←CHANGE T TO P IF PRODUCTION TAPE
//*
//SYSIN      DD    *
  COPY I=((JCLIN,R)),O=JCLOUT
/*
```

Figure 6: JCL to download Sample JCL Library

90.06 INSTALLING THE IAM ISPF DIALOG

INSTALLING THE PANELS AND MESSAGES LIBRARIES

While completely optional, customers are encouraged to install the IAM ISPF Dialog. The dialog includes some utility functions that may be quite useful for some quick problem diagnosis. They also provide panels to display information about IAM file activity in CICS regions and within IAMRLS. The IAM ISPF Dialog Panels and messages are contained in 2 libraries on the installation tape. To install these libraries, submit IAMSPFIN from the Installation Control Library after making the following changes as required for your environment:

1. Update the job card.
2. Change ISPPLIB='ISPF.PANEL.LIBRARY' to the name of the library you wish to contain the IAM panels.
3. Change ISPMLIB='ISPF.MESSAGES.LIBRARY' to the name of the library you wish to contain the IAM messages.
4. Change UNIT=TAPE to specify a tape drive capable of reading the product distribution tape that you received.

If you have installed the IAM panels and messages in libraries that are not specified in the ISPPLIB and ISPMLIB DD statements in the appropriate TSO logon procs, then you must add the dataset name specified for the panel library to the ISPPLIB concatenation, and add the dataset name specified for the message library to the ISPMLIB concatenation.

ADD THE IAM ISPF DIALOG TO AN ISPF OPTIONS MENU

You must add an option to the ISPF/PDF primary option menu (ISR@PRIM), or an option menu of your choice, to invoke the IAM ISPF DIALOG program. Add a line that describes the IAM ISPF DIALOG to the PANEL BODY.

Example:

% I +IAM – IAM Dataset Utilities

Add a line that selects the IAMISPF program to the lines that translate the user entered options into the appropriate panel or program name in the PROC section of the panel definition.

Example:

```
&ZSEL = TRANS( TRUNC (&ZCMD,'.')
0,'PANEL(ISPOPTA)'
1,'PGM(ISRBRO)'
.
other panel options
.
I,'PGM(IAMISPF)'            ← Add
.
other panel options
. , , ,
X,'EXIT'
* ,'?' )
```

ADD THE IAM LOAD LIBRARY TO THE TSO LOGON PROCS

If IAM has not been installed in a linklist library, you must make the IAMISPF load module, and other required IAM system modules, available under ISPF by adding the IAM load library to the STEPLIB DD statement in the appropriate logon procs. Please note that the use of the ISPLLIB DD for this purpose is strongly discouraged. If any of the IAM utility programs are executed from within IAMISPF via option U, **they will NOT be loaded from the ISPLLIB DD**. Additionally, some of the IAM utility functions require APF authorization which will require that all libraries specified in the STEPLIB DD be authorized.

90.06 CONTINUED . . .

APF AUTHORIZATION FOR THE IAM ISPF DIALOG

The IAMRECVR DIAGNOSE and PRINT functions available under option U.R utilize system facilities that require APF (Authorized Program Facility) authorization. The IAMXMONA program, which displays information about IAM files in CICS regions also requires APF authorization. If any of these functions are used from within the IAM ISPF DIALOG, you must provide authorization for those utility programs as discussed below. If you do not wish to execute these functions under the IAM ISPF DIALOG, you will not have to provide APF authorization for IAMRECVR.

If you have TSO/E release 2 or higher, you can provide APF authorization for IAMRECVR to run under the IAM ISPF DIALOG by updating the appropriate TSO authorized program table, as discussed below.

IEAAPFXX

The IAM load modules must be available to the IAM ISPF DIALOG via an APF authorized library.

The IAM load module library must be either:

(a) a linklist library that is APF authorized, or

(b) allocated to the STEPLIB DD in the LOGON PROC. Additionally, the IAM load module library and any other libraries that may be concatenated under the STEPLIB DD MUST be listed in the IEAAPFxx member of SYS1.PARMLIB even if they are also in the LNKLSTxx member. Please note that you can NOT use the ISPLLIB DD for this purpose.

TSO AUTHORIZED PROGRAM TABLE

Programs that are allowed to run with APF authorization under TSO must be listed in the TSO Authorized Program Table. As discussed below, you must add IAMRECVR and IAMXMONA to the appropriate table.

Starting with TSO/E release 4, the list of authorized programs may be specified in member IKJTSO00 in SYS1.PARMLIB. The old method of changing the table within a load module in the Link Pack Area is still supported. The authorized program lists are documented in the IBM manual "TSO/E Customization".

UPDATING IKJTSO00

If your installation uses the IKJTSO00 member, then add IAMRECVR and IAMXMONA to the AUTHTSF NAMES list. If not, then use the table below to determine the CSECT(s) and LOAD module to change. An IPL or PARMLIB command is required in order for the changes to IKJTSO00 to take effect. The CSECT name(s) and load module of the table that must be updated for various levels of MVS and TSO/E are as follows:

Operating System	TSO	Load Module	CSECT(s)
MVS/XA+	TSO/E R3+	IKJTABLS	IKJEFTE8 and IKJEFTAP
MVS/XA+	TSO/E R2.1	IKJTABLS	IKJEFTE8
MVS/XA+	TSO/E R2.0	IKJEFT02	IKJEFTE8

"+" means "or higher"

An IPL with CLPA is required for the updated TSO authorized program table to take effect. The authorized program tables are documented in the IBM manual “TSO/E Customization”.

90.07 INSTALLING IAM CICS MONITOR

IAMXMON consists of a single program and a single mapset:

Program: IAMXMON language (Assembler) EXECKEY(CICS)

Mapset: IAMXMAP

These modules must be copied from the IAM LOADLIB into a LOADLIB that is part of the DFHRPL concatenation in order to work properly. IAMXMON is a CICS application, not part of the IAM access method code. It must be defined with EXECKEY(CICS). A transaction ID must be assigned to invoke the IAMXMON program, such as IAMX or IMON and be defined with TASKDATAKEY(CICS).

The details of installing IAMXMON are provided below:

- Copy the modules IAMXMON and IAMXMAP into a load library that is in the DFHRPL concatenation. IAMXMON is the Execution Monitor program and IAMXMAP is the mapset.

JCL TO COPY IAMXMON INTO DFHRPL LIBRARY

```
//COPYIAMX EXEC  PGM=IEBCOPY,REGION=2M
//SYSPRINT  DD   SYSOUT=*
//SYSUT3    DD   UNIT=SYSDA,SPACE=(CYL,(2))
//SYSUT4    DD   UNIT=SYSDA,SPACE=(CYL,(2))
//IAMLIB    DD   DISP=SHR,DSN=iam.loadlib          ← User Change
//DFHRPL    DD   DISP=SHR,DSN=cics.rpl             ← User Change
//SYSIN     DD   *
  COPYMOD  INDD=((IAMLIB,R)),OUTDD=DFHRPL
  SELECT   MEMBER=IAMXMON
  SELECT   MEMBER=IAMXMAP
/*
```

Figure 7: Example of JCL to copy IAMXMON to DFHRPL

- Define new CICS Program Properties Table (PPT) entries for the Execution Monitor mapset which must be called IAMXMAP:

PPT ENTRY FOR MAPSET: IAMXMAP

```
OBJECT CHARACTERISTICS                          CICS RELEASE = 0410
CEDA  View Mapset( IAMXMAP   )

 Mapset          : IAMXMAP
 Group           : IAMAPP1
 Description     :
 REsident        : No                     No | Yes
 USAge           : Normal             Normal | Transient
 USElpacopy      : No                     No | Yes
 Status          : Enabled           Enabled | Disabled
 RSl             : 00                   0-24 | Public
```

Figure 8: Example of Defining PPT for the mapset: IAMXMAP

- Define a new CICS Program Properties Table (PPT) entry for the Execution Monitor program which must be call IAMXMON:

90.07 CONTINUED . . .

PPT ENTRY FOR PROGRAM: IAMXMON

```
OBJECT CHARACTERISTICS                    CICS RELEASE = 0410
 CEDA  View PROGram( IAMXMON  )
 PROGram         : IAMXMON
 Group           : IAMAPP1
 DEscription     :
 Language        : Assembler       CObol | Assembler | Le370 | C | Pli | Rpg
  RELoad         : No                 No | Yes
  RESident       : No                 No | Yes
  USAge          : Normal         Normal | Transient
  USElpacopy     : No                 No | Yes
  Status         : Enabled       Enabled | Disabled
  RSl            : 00               0-24 | Public
  Cedf           : Yes               Yes | No
  DAtalocation   : Below           Below | Any
  EXECKey        : Cics             User | Cics         ← Note 1
  REMOTE ATTRIBUTES
  REMOTESystem   :
  REMOTEName     :
 USElpacopy      : No                 No | Yes
 Transid         :
 EXECUtionset    : Fullapi       Fullapi | Dplsubset
```

Figure 9: Define PPT Entry for program: IAMXMON

Note 1: If CICS Storage Protection is active, the IAMXMON program **MUST be defined with EXECKEY(CICS).**

- Define a new CICS Program Control Table (PCT) entry defining the Transaction ID (TRANSID) you will use to invoke the IAM Execution Monitor. The selected TRANSID can be any unique 1 to 4 character identifier.

90.07 CONTINUED . . .

PCT ENTRY FOR IAMXMON TRANSACTION

```
OBJECT CHARACTERISTICS                         CICS RELEASE = 0410
 CEDA  View TRANSaction( IAMX )
 TRANSaction      : IAMX
 Group            : IAMAPP1
 DEscription      :
 PROGram          : IAMXMON
 TWasize          : 00000                  0-32767
 PROFile          : DFHCICST
 PArtitionset     :
 STAtus           : Enabled                Enabled | Disabled
 PRIMedsize       : 00000                  0-65520
 TASKDATALoc      : Below                    Below | Any
 TASKDATAKey      : Cics                      User | Cics  ← Note 2
 STOrageclear     : No                          No | Yes
 RUnaway          : System                  System | 0-2700000
 SHutdown         : Disabled              Disabled | Enabled
 ISolate          : Yes                        Yes | No
 REMOTE ATTRIBUTES
 DYnamic          : No                          No | Yes
 REMOTESystem     :
 REMOTEName       :
 TRProf           :
 Localq           :                             No | Yes
 SCHEDULING
 PRIOrity         : 001                    0-255
 TClass           : No                          No | 1-10
 TRANClass        : DFHTCL00
 ALIASES
 Alias            :
 TASKReq          :
 XTRanid          :
 TPName           :
 XTPname          :
 RECOVERY
 DTimout          : No                          No | 1-6800
 INdoubt          : Backout                Backout | Commit | Wait
 RESTart          : No                          No | Yes
 SPurge           : No                          No | Yes
 TPUrge           : No                          No | Yes
 DUmp             : Yes                        Yes | No
 TRACe            : Yes                        Yes | No
 SECURITY
 RESSec           : No                          No | Yes
 Cmdsec           : No                          No | Yes
 Extsec           : No
 TRANSec          : 01                        1-64
 RSl              : 00                        0-24 | Public
```

Figure 10: Define PCT Entry for IAMXMON Transaction

Note 2: If CICS Storage Protection is active, the transaction **MUST be defined with TASKDATAKEY(CICS).**

- Restart CICS, or INSTALL the transaction, program and map entries using CEDA.

90.10 SETTING UP IAM RLS (RECORD LEVEL SHARING)

IAM provides a facility to enable concurrently executing jobs and CICS regions to share IAM files for update with full data integrity, within a single MVS system image, referred to as IAM RLS (Record Level Sharing). IAM RLS provides for sharing IAM datasets, with serialization at the record level. This means that only one batch job or CICS transaction can have update authority for a specific record at any point in time. Once the update has completed and the transaction or job has either finished or reached a recovery syncpoint, then the record lock is released and the record can be subsequently updated by another job or transaction. If a job or CICS region abends while holding a record lock(s) for any datasets that are recoverable then the record locks are retained until a recovery process has occurred.

If you have any share option 3 or 4 IAM files, Innovation strongly recommends that you set up and run IAM RLS. If you have any VSAM files with share option 3 or 4 that you haven't converted to IAM, by activating IAM RLS you may now be able to convert those files.

IAM RLS accomplishes the sharing by providing a server address space, in which all I/O for shared IAM datasets are processed. IAM RLS provides journaling and recovery capabilities using the IAM RLS journals. There is some setup required to enable and activate IAM RLS processing. The major steps include the following:

- Setting up the IAM RLS procedure and parameter libraries
- Specification of various IAM RLS parameters
- Setting up dataset level automatic eligibility criteria
- Allocating IAM journal datasets, and setting up backup procedures for the journals

The information necessary on the IAM RLS procedure ("PROC") is included on the following pages. All of the other information on setting up IAM RLS is in Section 20, Using IAM Record Level Sharing.

Full Sysplex Record Level Sharing support will be provided in a subsequent release of IAM.

IAM RLS PROC IAM RLS runs as a started task in an MVS system. It has a number of differences from the IAMSTART procedure, which just installed some IAM modules into Dynamic LPA, and terminated. The IAMRLS procedure runs continuously as a non-swappable address space, providing services to the various jobs and CICS regions that are sharing IAM files. In general, IAM RLS will be a task that should be kept running all the time. However, IAM RLS can be shut down and restarted whenever it may be necessary to do so without impacting processing for IAM datasets that are not using IAM RLS.

Because IAM RLS is an I/O service provider, you will need to **make sure that the IAM RLS address space is provided with a high enough priority to service the CICS transactions and batch jobs to meet their response time requirements.** It probably should be given a priority equal to or higher than the most critical CICS region or batch job to insure that adequate service is provided. Specifying too low of a priority will result in poor response times for CICS transactions and elongated run times for batch jobs.

If you are installing IAM RLS, you can replace the execution of the IAMSTART procedure with the start up of IAM RLS. IAM RLS will automatically run IAMSTART to install the IAM VSAM Interface, if it is not already active. Note that subsequently shutting down IAM RLS will not deactivate the IAM VSAM interface, the IAM VSAM interface will be left active.

Normally you will only have one IAM RLS address space active at a time. To provide a testing capability for maintenance levels and new releases, you can also run a "test" IAM RLS address space, which will provide support for only the specified job names. The "test" IAM RLS must have a different name than the production version, and you can only share datasets that are not being used within the production IAM RLS.

90.10 CONTINUED . . .

IAMRLS PROC

In the ICL (Installation Control Library) that was copied from the product tape, is an example of the IAMRLS procedure, which is shown below. This example can be edited and copied into one of your system proclibs. You will need to change the dataset name of the IAM load library, and the dataset name of the card image PDS to be used for parameters and overrides. You may also desire to change the default member names for the IAMRLS parameters and for the IAMRLS override.

```
//IAMRLS    PROC MBR=RLSPARM1,OVR=RLSOVRID,TBL=DSNTB
//IEFPROC   EXEC PGM=IAMRLS,
//             REGION=0M,DPRTY=(15,15),TIME=1440
//STEPLIB    DD  DSN=your.iam.loadlib,DISP=SHR
//SYSLIB     DD  DSN=your.iam.loadlib,DISP=SHR
//SYSPRINT   DD  SYSOUT=X
//RLSLOGDD   DD  SYSOUT=X
//SYSABEND   DD  SYSOUT=X
//IAMINFO    DD  SYSOUT=X
//IPARMLIB   DD  DISP=SHR,DSN=your.iam.icl(&MBR)
//IAMOVRID   DD  DISP=SHR,DSN=your.iam.icl(&OVR)
//IAMDSNTB   DD  DISP=SHR,DSN=your.iam.icl(&TBL)
```

Figure 11: Sample of IAM RLS PROC

IAMRLS TEST PROC

Should a need arise to test PTF's that you have received, or to test an updated maintenance level of IAM, you can start a test version of the IAMRLS address space. This will allow you to limit use of that IAMRLS address space to specified jobs. Starting an IAMRLS for test will also cause a test version of the IAM VSAM Interface to be started. To start a test version, add a parameter value to the execute statement of started task JCL, or specify it on a start command. This works the same as the test versions of IAM worked in the past. A job name or a mask with a job name prefix can be used which will allow the specified job names to be processed by the IAMRLS address space. Up to four (4) different job name(s) can be specified. The test IAMRLS address space must have a different name than the production IAMRLS address space, which will mean setting up a different proc for the test version. Note that an IAM dataset that is being processed under the production IAMRLS address space will not be able to be opened under the test IAMRLS address space.

An example of a test IAMRLS proc is shown below. The test job names are specified by the JOB parameter when starting the IAMRLST procedure. The "test" IAM RLS will need to have a different parmlib member so that different journal datasets can be specified.

```
//IAMRLST   PROC MBR=RLSPARMT,OVR=RLSOVRID,TBL=DSNTB
//IEFPROC   EXEC PGM=IAMRLS,
//             PARM='TEST,JOBNAME=(&JOB)',
//             REGION=0M,DPRTY=(15,15),TIME=1440
//STEPLIB    DD  DSN=your.iam.loadlib,DISP=SHR
//SYSLIB     DD  DSN=your.iam.loadlib,DISP=SHR
//SYSPRINT   DD  SYSOUT=X
//RLSLOGDD   DD  SYSOUT=X
//SYSABEND   DD  SYSOUT=X
//IAMINFO    DD  SYSOUT=X
//IPARMLIB   DD  DISP=SHR,DSN=your.iam.icl(&MBR)
//IAMOVRID   DD  DISP=SHR,DSN=your.iam.icl(&OVR)
//IAMDSNTB   DD  DISP=SHR,DSN=your.iam.icl(&TBL)
```

Figure 12: Sample of Test IAM RLS PROC

90.10 CONTINUED . . .

IAMRLS DD STATEMENTS The DD statements needed in the IAMRLS proc are described below.

DDNAME	Description
STEPLIB	Specifies the IAM load library dataset. This DD is optional if IAM is in the system link list.
SYSLIB	Required DD statement that specifies the IAM load library dataset.
SYSPRINT	Required DD statement, specifying a sequential output dataset that is normally printed. If specified as SYSOUT, make sure the class used is one that is not automatically purged, in case the output is needed for problem diagnosis.
RLSLOGDD	Required DD statement, specifying a sequential output dataset that is normally printed. This file contains a log of activity and error messages. If specified as a SYSOUT, make sure that the class used is one that is not automatically purged, in case the output is needed for problem diagnosis.
SYSABEND	An optional, but highly recommended DD statement, that will be used when a dump is taken due to an abend condition. An alternative would be to use a SYSMDUMP DD statement. Use of SYSUDUMP is not recommended, because it will contain insufficient information for problem diagnosis.
IAMINFO	An optional, but highly recommended DD statement that will contain activity reports for each IAM file that is produced whenever that file is closed. If specified as SYSOUT, make sure the class used is one that is not automatically purged, in case the output is needed for problem diagnosis. An alternative to using the printed output is to enable the IAM SMF record, which will save a machine-readable format of the same data produced on the IAMINFO report. The IAMINFO reports can then be obtained from the systems SMF data using the IAMSMF program.
IPARMLIB	An optional DD statement, which specifies an 80-character card image input dataset containing various parameters and options for IAMRLS. This dataset is typically a PDS.
IAMOVRID	An optional DD statement that specifies an 80-character card image input dataset, which contains the IAM override statements. This provides a mechanism for indicating parameters, such as number enabling journaling for all files processed by IAM RLS.
IAMDSNTB	An optional DD statement that specifies an 80-character card image input dataset. This dataset contains the dataset selection and dataset exclusion lists for IAMRLS processing.
ABNLIGNR	A DD DUMMY to bypass Abend-Aid dumps.
ABNLDUMP	A DD DUMMY to bypass ABEND-Aid dumps.

90.20 ACTIVATING THE IAM VSAM INTERFACE

IAM provides a system level VSAM interface that provides the capability to use IAM datasets in place of VSAM KSDS or ESDS files generally without any JCL or program changes. To accomplish this, IAM must be activated after each IPL. The activation procedure runs as a batch job or started task that terminates after initializing the IAM interface. Alternatively, if you are using IAM RLS, it will automatically activate the IAM VSAM interface during initialization. This activation effectively loads the IAM VSAM interface modules into the Pageable Link Pack Area (PLPA) area of virtual storage. The IAM VSAM interface modules act as a front end to various VSAM services, such as Open and Close. If the dataset being processed is an IAM dataset, then the interface gives control to the appropriate IAM routines. If the dataset is not an IAM dataset then control is directed to the normal VSAM processing routine. Once a dataset is opened, I/O requests are handled directly by the access method responsible for the dataset, either IAM or VSAM. The IAM VSAM interface causes no interference or overhead with the processing of I/O requests to real VSAM files.

- **IAM IS NOT A REPLACEMENT FOR VSAM IN YOUR SYSTEM.**
- **IAM DOES NOT AFFECT THE NORMAL USE OF VSAM IN THE SYSTEM.**
- **VSAM CATALOG MANAGEMENT IS NOT CHANGED.**

COEXISTENCE WITH OTHER SOFTWARE PRODUCTS

There are wide varieties of other software products that also intercept the various services done for VSAM datasets. IAM can coexist in systems with these other software products installed. Depending on the function of the product and the various interfaces used, there may be a start order dependency between the different products to insure that the expected processing by each product occurs as anticipated. In general, if a product provides services and capabilities not applicable to IAM datasets, then they should be activated before IAM. Products that can be used for IAM datasets should be started after IAM. For example, VSAM buffering and performance products should be started before IAM. Other products, such as VSAM recovery and journaling software or VSAM space management packages that can be used for IAM datasets should be started after IAM.

There are some software products which require a specific setting in the IAM Global Options Table. These are identified in Section 90.04 of this manual and are shown in the table below.

Should you have any concerns or questions about IAM coexistence with other software products, contact the IAM technical support department at Innovation Data Processing.

90.20 CONTINUED . . .

START ORDER OF IAM The start order recommendations of various products as they relate to IAM along with an indication of there being an IAM Global Option setting required are shown in the table below. Care must also be taken to insure that the various products are not starting at the same time because not all products follow the required locking protocol, and for some of the control block updates, there is no established serialization protocol. The order of the various start commands in the SYS1.PARMLIB member COMMNDxx is no guarantee of the order in which the product will actually perform it's initialization since the commands are executed concurrently.

Product	Start Order	Global Option
BIM Products	Before IAM	ENABLE=BIM
CA-DISK	Before IAM	n/a
HYPERCACHE	Before IAM	n/a
BMC MAINVIEW		DYNCAT=YES
PROSMS	Before IAM	n/a
Recovery Plus	After IAM	n/a
CA-Allocate (SAMS (VAM))	After IAM	ENABLE=VAM
SmartBatch (IBM)	Before IAM	n/a
Transaction Server	After IAM	LOADABWO=YES

Figure 13: VIF Start Order with Other Software Products

The best way to insure the proper start order is by the use of automated operations software. Such software can determine that a process is complete by such means as checking for a specific WTO message, before initiating another process. For IAM the message to check for is the ***IAMW81 THE IAM SYSTEM MODULES ARE ACTIVE.***

Another alternative is to start a multiple step PROC, which includes the IAMSTART procedure at the appropriate place. This will work providing that the steps prior to the IAMSTART terminate after the product initialization, as IAMSTART does.

ACTIVATING IAM The IAM Installation Control Library contains six (6) members to control the IAM VSAM Interface in your system. If you are running IAM RLS, you can let IAM RLS activate the IAM VSAM interface, instead of using the IAMSTART proc.

MEMBER NAME	DESCRIPTION
VIFSTART	A job stream to activate IAM.
VIFSTATS	A job stream to check the status of IAM in the system.
VIFSTOP	A job stream to deactivate IAM.
VIFTEST	A job stream to activate a TEST IAM.
IAMSTART	A procedure that activates IAM.
IAMRLS	A procedure that starts IAM RLS and activates IAM(See Sec. 90.11)
IAMCOMM	A command to activate IAM at IPL time.

Figure 14: ICL Members to Control the IAM VSAM Interface

90.20 CONTINUED . . .

VIFSTART JOB The VIFSTART job can be used to activate the IAM VSAM Interface. This job requires both a STEPLIB and a SYSLIB DD statement, which specify the APF authorized IAM load library. If you already have a version of IAM active, refer to the section on Concurrent IAM and Test IAM below before starting the new version. Before executing this job, if you have any of the software products that are to be started after the IAM that are currently active, shut them down temporarily before starting IAM. After starting IAM, those software products can be restarted.

```
//VIFSTART JOB (IAM),'IAM-VIF'
//******
//*    THIS JOB HAS BEEN CREATED FOR IAM V8.0
//*    IT ACTIVATES THE IAM SYSTEM LEVEL VSAM INTERFACE
//*    THE PURPOSE OF THIS JOB IS TO INSTALL THE IAM SYSTEM
//*    LEVEL VSAM INTERFACE IN AN OS/390 or z/OS SYSTEM.
//*    USER CHANGES:
//*      CHANGE 'USER.LIB' TO NAME OF IAM LOAD LIBRARY
//******
//START     EXEC PGM=IAMSTART
//STEPLIB    DD  DISP=SHR,DSN=USER.LIB               ← USER CHANGE
//SYSLIB     DD  DISP=SHR,DSN=USER.LIB               ← USER CHANGE
//SYSUDUMP   DD  SYSOUT=*
//
```

Figure 15: Example of the VIFSTART Job from the ICL

VIFSTATS JOB The VIFSTATS job can be used to verify the status of the IAM VSAM Interface on your system. If you have multiple versions of IAM, always run the VIFSTATS with a STEPLIB to the most recent version. This is because older versions of IAM may not be able to properly display information on the newer versions. This job will provide WTO messages indicating the status of all levels of the IAM VSAM Interface on your system. The JOBLOG also includes messages indicating the names of the various modules IAM has placed in the PLPA, along with their entry point and level. If you have installed the IAM ISPF panels, this same information can be displayed on your terminal using the U.V options from the IAM primary panel.

```
//VIFSTATS JOB (IAM),'IAM-VIF'
//******
//*    THIS JOB HAS BEEN CREATED FOR IAM V8.0
//*    IT REPORTS ON THE STATUS OF THE IAM SYSTEM LEVEL
//*    VSAM INTERFACE.
//*    USER CHANGES:
//*      CHANGE 'USER.LIB' TO NAME OF IAM LOAD LIBRARY
//******
//STATS     EXEC PGM=IAMSTATS
//STEPLIB    DD  DISP=SHR,DSN=USER.LIB             ← USER CHANGE
//SYSLIB     DD  DISP=SHR,DSN=USER.LIB             ← USER CHANGE
//SYSUDUMP   DD  SYSOUT=*
//
```

Figure 16: Example of VIFSTATS Job from the IAM ICL

90.20 CONTINUED . . .

DEACTIVATING IAM

The IAM VSAM Interface can be deactivated with the VIFSTOP job. This job will not remove the IAM VSAM Interface modules from the PLPA, but will deactivate them. Should you absolutely need to remove the IAM modules from PLPA without doing an IPL, contact Innovation Data Processing for the procedure to follow.

NOTE: To reactivate IAM use the VIFSTART procedure.

WARNING: If you deactivate IAM in a production environment, all OPEN, CLOSE and IDCAMS processing against IAM files will fail.

VIFSTOP JOB

```
//VIFSTOP JOB (IAM),'IAM-VIF'
//******
//*    THIS JOB HAS BEEN CREATED FOR IAM V8.0
//*    IT DEACTIVATE THE IAM SYSTEM LEVEL VSAM INTERFACE
//*    USER CHANGES:
//*       CHANGE 'USER.LIB' TO NAME OF IAM LOAD LIBRARY
//******
//STOP       EXEC PGM=IAMPARE
//STEPLIB     DD  DISP=SHR,DSN=USER.LIB          ← USER CHANGE
//SYSLIB      DD  DISP=SHR,DSN=USER.LIB          ← USER CHANGE
//SYSUDUMP    DD  SYSOUT=*
//
```

Figure 17: Example of VIFSTOP Job from the ICL

IAMSTART PROC

Also provided in the ICL is an example of a PROC that can be copied into a system PROCLIB, so that the IAM VSAM Interface can be automatically or manually started by the operator START command. This PROC is set up with one operand, LIB= that can be used to override the default STEPLIB and SYSLIB DD statements, which are required to be in the PROC. The STEPLIB can be removed if the PROC is only used to start the version of IAM that is in the system LINKLIST. If you have a need to be able to activate IAM without any DD cards, contact Innovation Data Processing for custom zap C-80.0009 to enable IAM starting directly from the LINKLIST if no SYSLIB DD is allocated.

```
//IAMSTART  PROC  LIB='USER.LIB'          ←USER CHANGE
//* MEMBER(IAMSTART)
//******
//*    THIS PROCEDURE HAS BEEN CREATED FOR IAM V8.0
//*    IT ACTIVATES THE IAM SYSTEM LEVEL VSAM INTERFACE.
//*    THE PURPOSE OF THIS PROCEDURE IS TO PROVIDE A MEMBER THAT
//*    CAN BE PLACED IN SYS1.PROCLIB THAT WILL ACTIVATE THE
//*    INTERFACE WHEN IT IS INVOKED BY A SYSTEM START COMMAND.
//*    USER CHANGES:
//*       CHANGE 'USER.LIB' TO NAME OF IAM LOAD LIBRARY
//******
//******
//START      EXEC PGM=IAMSTART
//STEPLIB     DD  DISP=SHR,DSN=&LIB
//SYSLIB      DD  DISP=SHR,DSN=&LIB
//SYSUDUMP    DD  SYSOUT=*
```

Figure 18: Example of IAMSTART PROC from the IAM ICL

90.20 CONTINUED . . .

ACTIVATING IAM AT IPL TIME

After you have completed your testing, or possibly before, you will want to start IAM automatically at each IPL. To have the IAM automatically activated with each IPL, use the following procedure:

1. In member IAMSTART of the Installation Control Library, change the library name specified in the 'LIB=' statement to reflect your IAM load library.
2. Copy the member 'IAMSTART' into SYS1.PROCLIB (or the procedure library used by your installation for system level user procs). This is the start up procedure for IAM.
3. Copy the member IAMCOMM from the IAM Installation Control Library into the SYS1.PARMLIB member COMMND00 (or the COMMNDxx member used by your installation). This will create a record that will result in a system start command being issued for the IAM start up procedure at IPL time. The record in that member should look like the following statement.

COM='START IAMSTART'

You have now completed the installation of a dynamic IAM system level VSAM interface that will automatically be activated each time the operating system is IPL'd.

TESTING IAM V8.0 WITH PRIOR VERSIONS OF IAM ACTIVE

If you are a current customer of IAM, you can install and test new versions of IAM without affecting any of your existing jobs. Programs can continue to use the prior version(s) until you choose to have them use the new version. Follow the appropriate testing procedure described below. Normally, your production version of IAM will be in a LINKLIST library. The new test version should be installed in a separate APF authorized library. If the library containing the new version is in the LINKLIST, it must be after the current version while testing is in progress.

One consideration with testing multiple levels of IAM is that the only version that can be used to define IAM datasets in JCL is the version of IAM that is in the LINKLIST. This is because the JOBLIB or STEPLIB specified in the JCL for the job to be executed is not used by the system initiator when it is performing allocation of the dataset.

RUNNING MULTIPLE IAM

If you are installing a new version of IAM, such as Version 8.0 while Version 7.0 is your production IAM, you can run multiple levels of IAM for your testing. The major rule to be followed is that the oldest version of IAM must be started first. There are two different ways to run multiple levels of IAM. The first way is called Concurrent IAM, and is generally used when testing new versions of the product. The second way is called Test IAM, and is normally used when testing different maintenance levels of the same version of IAM. When running multiple versions of IAM, the primary production version matches the version of IAM that is in the system LINKLIST. The other versions are accessed through a JOBLIB or STEPLIB pointing to the alternate IAM load library.

CONCURRENT IAM

Concurrent IAM works by comparing the loaded IAM module level with the level of the IAM VSAM interface using an internal level identifier for the IAM VSAM interface. When a VSAM function is intercepted by IAM, such as DEFINE, LISTCAT, OPEN, or CLOSE, IAM will load the appropriate IAM routine to process the request. If the internal level in the loaded IAM routine matches the IAM currently processing the request, then processing will proceed. If the internal level is not the same, then the current IAM percolates control to the prior level of IAM. If no match is found, VSAM will attempt to process the file, which will most likely result in a failure.

The Concurrent IAM VSAM can be started using the VIFSTART job or the IAMSTART PROC pointing to the new IAM load library. The one restriction is that the oldest version of IAM must be started first, and the newest version started last. Control over which version of IAM will be used is then based on the JOBLIB or STEPLIB being used by the job step. With Concurrent IAM, there is no concern about the job names, as there is with the Test IAM procedure.

90.20 CONTINUED . . .

TEST IAM IAM provides the capability to have a TEST version of the IAM VSAM Interface active. The TEST version can be used to test new versions and new maintenance levels of IAM, while having prior versions or maintenance levels of the IAM active. The TEST IAM interface utilizes a jobname screening mechanism to determine eligibility for using the TEST version. When activating a TEST version of IAM, a PARM of TEST is specified, followed by JOBNAME=jobname. The jobname must match the specified value to utilize the TEST version. The jobs that are utilizing the TEST version of IAM **MUST HAVE A JOBLIB OR STEPLIB** to the new IAM load library. A jobname prefix mask can be specified, by coding the prefix followed by an *. For example, specifying JOBNAME=myid*, all jobs that begin with the literal "myid" will be qualified for the test version. Up to five different jobnames or jobname groups can be specified. Multiple job names are specified by enclosing the names within parenthesis. The jobname(s) can be easily modified by rerunning the startup of the TEST version with a different value(s) for the JOBNAME parameter. Specifying JOBNAME=* will make all jobs eligible for the TEST version.

VIFTEST JOB In the IAM ICL, there is an example job stream to start a TEST version of IAM, as member VIFTEST. The VIFTEST JCL is as follows:

```
//VIFSTEST JOB (IAM),'IAM-VIF'
//******
//*    THIS JOB HAS BEEN CREATED FOR IAM V8.0
//*    IT ACTIVATES THE IAM SYSTEM LEVEL VSAM INTERFACE
//*    THE PURPOSE OF THIS JOB IS TO INSTALL THE IAM SYSTEM
//*    LEVEL VSAM INTERFACE IN AN MVS OR OS/390 SYSTEM
//*    FOR TESTING PURPOSES.
//*    USER CHANGES:
//*      CHANGE 'USER.LIB' TO NAME OF IAM LOAD LIBRARY
//*                JOBNAME=? TO JOBNAME=
//*                          THE JOBNAME TO BE TESTED, OR
//*                          TO [PREFIX]* FOR JOB NAME PREFIX, OR
//*                          TO * FOR ALL JOBS, OR
//*      JOBNAME=(jobname1,jobname2,...,jobname5) FOR
//*      MULTIPLE JOBNAMES
//******
//STARTEST  EXEC PGM=IAMSTART,PARM='TEST,JOBNAME=?'
//STEPLIB    DD  DISP=SHR,DSN=USER.LIB               ← USER CHANGE
//SYSLIB     DD  DISP=SHR,DSN=USER.LIB               ← USER CHANGE
//SYSUDUMP   DD  SYSOUT=*
//
```

Figure 19: Example of Starting a TEST VIF

As stated above, to change the name of the job(s) eligible for the TEST IAM version, simply rerun the above job with a new value specified for JOBNAME=.

DEACTIVATING TEST IAM To deactivate the TEST IAM version, use the following JCL:

```
//VIFSTOPT  JOB (IAM),'IAM-VIF'
//STARTEST  EXEC PGM=IAMPARE,PARM='TEST'
//STEPLIB    DD  DISP=SHR,DSN=USER.LIB               ← USER CHANGE
//SYSLIB     DD  DISP=SHR,DSN=USER.LIB               ← USER CHANGE
//SYSUDUMP   DD  SYSOUT=*
//
```

Figure 20: Example of JCL to Stop a Test VIF

90.20 CONTINUED . . .

TEST IAM CONSIDERATIONS

There are a few considerations when using a TEST IAM. The TEST IAM capability merely screens which job(s) are eligible for being processed by the new IAM. To be a valid test, the job must have access to the new IAM load modules via JOBLIB or a STEPLIB. The TEST IAM does not have the capability to dynamically select or force the library from which the IAM modules are loaded. This can cause failures or unexpected results if this capability is not appropriately used.

If the other version(s) of IAM that are active have different internal level values, then the TEST IAM behaves like a Concurrent IAM, with the exception that only those jobs whose name matches the specified JOBNAME value will be considered for the TEST IAM. For example, if a job has a JOBLIB or STEPLIB with the new IAM load library, but does not have a jobname match, it will end up processing with the real VSAM code, which will most likely cause an IEC161I error message. However, if there is a jobname match but the IAM modules that are accessed by that job are from the prior version, then the prior version of IAM will be used.

When using a TEST IAM to test a new maintenance level, where the internal level values are the same, the TEST IAM controls which version of the IAM will handle the job. For job(s) whose name matches the specified value, they will be handled by the new IAM. Job(s) whose names do not match the specified value will be processed by the other (production) IAM. It is the users responsibility to insure that the correct version of the IAM libraries are used by the TEST and non-TEST jobs. This is easily controlled if the production version is in the system LINKLIST, and the TEST version is accessed by JOBLIB or STEPLIB. If the test job name matches the job name specified for the TEST IAM, and the job has a JOBLIB or STEPLIB for the new library, it will be processed entirely by the new version. If a job name does match the TEST VIF parameter, but the job does not have the test JOBLIB or STEPLIB, it will then execute using the new VIF modules, but the old processing modules. If a job name does not match the TEST IAM parameter, but has the new JOBLIB or STEPLIB, it will execute using the old IAM VSAM Interface modules and the new processing modules. Either one of these sequences may cause an invalid test, or even fail to properly execute.

ACTIVATING MULTIPLE IAM DURING IPL

When testing a new version or maintenance level of IAM, it will generally be desirable to have the multiple levels of IAM started during IPL. This can be most easily accomplished by modifying the IAMSTART PROC to have multiple steps, each one starting a different version of IAM. Remember that the older versions of IAM must be started before the newer versions. Below is an example of the IAMSTART PROC modified to start up two concurrent versions of the IAM.

```
//IAMSTART  PROC  LIB='SYS1.IAM70.LOADLIB'
//* MEMBER(IAMSTART)
//******
//*    THIS PROCEDURE HAS BEEN CREATED TO START MULTIPLE
//*    VERSIONS OF THE IAM VIF.
//*    USER CHANGES:
//*      CHANGE 'SYS1.IAM70.LOADLIB' TO NAME OF PRODUCTION IAM
//*      CHANGE 'SYS1.IAM80.LOADLIB' TO NAME OF TEST IAM
//******
//******
//START70   EXEC  PGM=IAMSTART
//STEPLIB    DD   DISP=SHR,DSN=&LIB
//SYSLIB     DD   DISP=SHR,DSN=&LIB
//SYSUDUMP   DD   SYSOUT=*
//START80   EXEC  PGM=IAMSTART
//STEPLIB    DD   DISP=SHR,DSN=SYS1.IAM80.LOADLIB
//SYSLIB     DD   DISP=SHR,DSN=SYS1.IAM80.LOADLIB
//STATS     EXEC  PGM=IAMSTATS
//STEPLIB    DD   DISP=SHR,DSN=SYS1.IAM80.LOADLIB
//SYSLIB     DD   DISP=SHR,DSN=SYS1.IAM80.LOADLIB
```

Figure 21: Starting Multiple IAM Versions during IPL

90.20 CONTINUED . . .

ACTIVATING MULTIPLE IAM VERSIONS WITH A TEST IAM

You can also start up multiple levels of IAM including one as a TEST IAM. Below is an example of a PROC to do that. Remember that if you need to change the value of the JOBNAME all you need to do is to rerun the test IAM start with different jobname(s) specified.

```
//IAMSTART  PROC  LIB='SYS1.IAM70.LOADLIB'
//* MEMBER(IAMSTART)
//******
//*    THIS PROCEDURE HAS BEEN CREATED TO START MULTIPLE
//*    VERSIONS OF THE IAM VIF WITH ONE AS A TEST VERSION.
//*    USER CHANGES:
//*      CHANGE 'SYS1.IAM70.LOADLIB' TO NAME OF PRODUCTION IAM
//*      CHANGE 'SYS1.IAM80.LOADLIB' TO NAME OF TEST IAM
//*      CHANGE 'iamtst*' TO THE DESIRED JOB NAME PREFIX FOR THE
//*      TEST VIF.
//******
//******
//IAMPROD   EXEC  PGM=IAMSTART
//STEPLIB    DD   DISP=SHR,DSN=&LIB
//SYSLIB     DD   DISP=SHR,DSN=&LIB
//SYSUDUMP   DD   SYSOUT=*
//IAMTEST   EXEC  PGM=IAMSTART,
//                PARM='TEST,JOBNAME=iamtst*'
//STEPLIB    DD   DISP=SHR,DSN=SYS1.IAM80.LOADLIB
//SYSLIB     DD   DISP=SHR,DSN=SYS1.IAM80.LOADLIB
//STATS     EXEC  PGM=IAMSTATS
//STEPLIB    DD   DISP=SHR,DSN=SYS1.IAM80.LOADLIB
//SYSLIB     DD   DISP=SHR,DSN=SYS1.IAM80.LOADLIB
```

Figure 22: Example of PROC to Start Multiple IAM Versions with one as a TEST IAM

90.30 TESTING IAM

Once you have completed the above steps, including activating IAM you are now ready to test IAM.

INSTALLATION VERIFICATION TEST

To make a quick test to see if IAM is properly installed in the system submit the job 'TESTVIF' from the IAM Installation Control Library. This job will execute IDCAMS to DEFINE, REPRO and DELETE an IAM file. A LISTCAT is executed against the IAM file. If IAM is properly installed, LISTCAT will show the file as a non-VSAM file. The IAMPRINT DD statement will display the characteristics of the IAM file. You should also verify from the IAMINFO and IAMPRINT reports that the any of the changed Global Options did in fact take effect.

If LISTCAT shows the file as VSAM, then the IAM VSAM Interface may not be properly installed. Review the Installation Instructions to make sure that all required tasks have been completed. If there are any questions, contact IAM Technical Support at Innovation Data Processing.

NOTE: Before submitting the test job, review the JCL and control statements, making whatever changes are appropriate.

SMF ANALYSIS

You are now ready to test IAM in your system. If you are a new user, the first thing you should do is identify your most active VSAM clusters. To aid you in this function, IAM supplies you with a SMF Analysis program (IAMSMFVS). This program is documented in Section 40.01 of this manual. To use this program you must be collecting SMF type 64 and either SMF type 4 or 30 subtype 4 records.

A sample procedure is supplied as member 'IAMSMFVS' in the IAM Installation Control Library.

Review this procedure making any appropriate changes. You can select from your current SMF dataset or from history SMF tapes. It is recommended that you scan from one week to one month's worth of SMF history to get a true picture of your VSAM usage. Submit member 'IAMSMFVS' to display the VSAM files with the most activity in your system. The first report will display the top 100 VSAM files in descending EXCP order. The second report will display the VSAM files in dataset name sequence. Converting just a few of your most active VSAM files can give noticeable and immediate benefits to your CICS and batch systems.

NOTE: If IAMSMFVS gives a 'IAM495 NO RECORDS MATCHED SELECTION CRITERIA' message, it probably means that you are not collecting SMF type 64 records. Check SYS1.PARMLIB member SMFPRMxx to see what records SMF is collecting. If type 64 records are not collected, you must change this member. You must issue a SET SMF=xx command to activate the SMF change or wait for your next IPL. After you have collected at least one day's worth of data, run the IAMSMFVS program again.

GENERAL TESTING

Once you choose which VSAM files you wish to test with, use IDCAMS to DEFINE the IAM file(s). Take your existing IDCAMS procedure and add the parameter OWNER($IAM) to the DEFINE statement.

Next, use IDCAMS REPRO to copy the VSAM cluster to the IAM file. Run a LISTCAT ALL against the IAM file with an IAMPRINT DD statement specified. The IAMPRINT report will give you the number of tracks used by IAM in addition to all of the IAM characteristics. IAM will automatically release a portion, based on CA% Freespace, of the unused space allocated, if secondary allocation has been specified. Compare this to the VSAM cluster. Unfortunately, it is not easy to tell how much space VSAM really used. If the Data Component HI-USED-RBA is close to HI-ALLOC-RBA, the VSAM file is using most of its allocated space.

If you wish to calculate the space used by the VSAM cluster use the following calculation. Use the values found in the DATA component. Divide the HI-USED-RBA by CISIZE. Divide the result by CIs per CA. Multiply this value by the number of tracks per CA. This will yield the number of tracks used by the Data Component and imbedded index (if specified). Add in the tracks for the Index. This will give you the approximate number of tracks used by the VSAM cluster.

One of the easiest tests you can make is to compare an IDCAMS REPRO of a VSAM cluster and IAM file to tape. Many applications use REPRO to make backup copies of their VSAM files. When the jobs are completed, compare the wall clock time, CPU time (TCB and SRB) and EXCPs issued by each job. This will give you a comparison of the resources used to sequentially read an entire VSAM cluster versus using an IAM file.

90.30 CONTINUED . . .

PARALLEL TESTS

You are now ready to run parallel tests of jobs using VSAM clusters compared to IAM file(s). Select an application to test, using the IAMSMFVS report as a guide. Use IDCAMS to DEFINE the IAM file specifying a different cluster name(s) then REPRO the VSAM cluster(s) into the IAM file(s). Use a copy of any file(s) that may be changed by the application. Run the production job against the VSAM cluster(s). Use the same JCL for the parallel run, changing the JCL to point to the new IAM file(s). Insert a STEPLIB pointing to the IAM load library if it is not in the LINKLIST. Re-execute the production run using the IAM file(s). Compare the results from each run. If you do not have the reporting tools necessary to get the SMF data you require, use the IAM supplied program IAMSMF (Section 41.01). Compare the statistics for each job. You will want to measure the wall clock time, CPU time (SRB and TCB), and EXCP counts (DATA and INDEX for VSAM). In addition, you will want to compare the disk space used by IAM versus the VSAM clusters.

IAMINFO DD STATEMENT

Add the following statement to each of the steps using IAM data test.

```
//IAMINFO  DD  SYSOUT=*
```

If this statement is present IAM will print a Run Time Statistics report each time an IAM file is closed. The report will display the characteristics of the file, run time statistics (memory use, read and write I/Os, etc.), count of each command issued (GETS, PUTS, ADDS, DELETES, POINTS, etc.), number of buffers used and additional information. There is very little overhead associated with the report because all the statistics are kept regardless of whether or not an IAMINFO DD card is present. The information from the IAMINFO report can be very valuable to you and Innovation in determining what each job is doing. (see Section 10.70 IAM Reports for details on using this report.)

TESTING A NEW VERSION OF IAM

Testing a new version of IAM or a new maintenance level of IAM is straight forward. If you have activated a Concurrent IAM VSAM interface, then just add the appropriate STEPLIB or JOBLIB to the job steps or jobs that you want to run as a test with the new version or level of IAM. If you are using a TEST IAM, then the job name(s) must match the value specified when you last started the TEST IAM along with the appropriate STEPLIB or JOBLIB. By having IAMINFO reports and / or IAMPRINT reports, you can verify that the job is running with the new version or level from the heading on those reports.

If you plan on using the same IAM datasets with your old version of IAM, make sure that you do not select Global Options or IAM Overrides that may cause the dataset to be unusable under the prior version. Please note that Version 8 supports hardware compression and DFSMS extended format datasets, which are not supported by the prior releases.

90.31 IAM INSTALLATION QUESTIONS

QUESTIONS REGARDING INSTALLATION

Here are the answers to some common questions that may be encountered after the installation of IAM.

Question: Why do I get a VSAM cluster when I have placed the parameter OWNER($IAM) in the IDCAMS DEFINE?

Answer: There are several possibilities:

1. The IAM VSAM Interface is not active in this system. Run the procedure VIFSTATS in the Installation Control Library to see if IAM is active. If IAM is not active, run VIFSTART to activate VIF.
2. The IAM load library is not in the LINKLIST. The JCL LOG in this case may indicate a S806 abend for an IAM module name. If you do not wish to put IAM in the LINKLIST, insert a STEPLIB pointing to the IAM library in the IDCAMS DEFINE job.
3. The IAM library is not authorized. The IAM library must be authorized.

Question: Why do I get VSAM error message IEC161I opening an IAM file?

Answer: There are several possibilities:

1. The IAM VSAM Interface is not active in this system. Run the procedure VIFSTATS in the Installation Control Library to see if IAM is active. If IAM is not active run VIFSTART to activate VIF.
2. The IAM load library is not in the LINKLIST. If you do not wish to put IAM in the LINKLIST, insert a STEPLIB pointing to the IAM library in the job referencing the IAM file.
3. The IAM library is not authorized. The IAM library must be authorized.

Question: Why might a file I converted to IAM all of a sudden go back to a VSAM cluster?

Answer: Some of the possibilities are:

1. Make sure that the IAM VSAM interface is active on the system.
2. There was a job that deleted and re-defined the file, but did not have the required JOBLIB or STEPLIB.
3. The JOBLIB or STEPLIB was not authorized in the system's APF list.
4. There was a job executed against the IAM file which DELETEd and re-DEFINEd the file as VSAM without the OWNER($IAM) parameter specified. You must find this job and change the DEFINE procedure. Use the IAMSMF program copy and query commands to report on all the jobs that have defined the cluster so you can identify the IDCAMS job causing the problem.

Question: Can I test a new version of IAM if I am already using an older version of IAM?

Answer: Yes, just insure the newest version of IAM's VIF is started after all prior versions of IAM are already active in the system. Normally, your jobs will require a JOBLIB or STEPLIB to the new version of IAM, until it is placed into production. Review detailed instructions in section 90.20.

Question: Can I use IAM with other software products that do VSAM allocation control or otherwise intercept VSAM?

Answer: Yes, IAM supports or coexists with the majority of the many software products that make decisions based on a file's DSORG and or interface with the same MVS services as IAM. The main consideration when using IAM in conjunction with one of these decision making facilities, especially one that requires a system interface or started task, is the order in which the product interfaces are activated. In general, if a program provides a VSAM only service such as altering VSAM buffering or activating LSR it should be started before IAM. However, if a product is applicable to IAM files, for example this would include but not be limited to CICS and batch VSAM journaling facilities like Recover Plus from BMC as well as VSAM allocation control products like CA-Allocate (formerly SAMS or VAM from Sterling Software), then IAM should be activated first.

90.40 MOVING A NEW VERSION OF IAM INTO PRODUCTION

OVERVIEW Several ways can be used to place the new version or maintenance release of IAM into production. In all cases, it will make for a cleaner and safer process if you can schedule an IPL. However, many installations are not able to do that, so alternative procedures are presented that will accomplish activating the new version or maintenance release as the level used by all of the jobs in the system without an IPL. Be sure to review the IPL implementation procedure because there are tasks you will need to do so that the new version of IAM will remain in place after an IPL occurs.

It is highly recommended that no matter what procedure you decide to follow, that you keep a copy of your prior version of IAM available in an authorized library. By doing so, you will have quick access to the prior version if it becomes necessary. The recommended procedures provide the capability of an emergency fall back to the prior version without an IPL, should unexpected problems occur. Once you have reached a level where the fall back is no longer necessary, you will be able to remove the old version of IAM from your system.

IMPLEMENTATION STRATEGY During the testing phase of the new level of IAM, you have been running with two versions of IAM active, using JOBLIB or STEPLIB to access the new version, while running out of LINKLIST for the production version. The recommended strategy is to migrate to the new level of IAM in two phases. During the first production phase for the new version, there will still be two levels of IAM started. However, the new level of IAM will be in placed in the LINKLIST, while the prior version of IAM will be accessible, if necessary, through a JOBLIB or STEPLIB. The old version of IAM, while started, will be deactivated. This will leave the prior level of IAM modules in PLPA, however they will be inactive. The old level of IAM will be ready to be reactivated if the need arises.

Once the new version has been completely accepted, then the second phase can be performed. This phase will remove the old level of IAM from the system completely. During this phase, changes will be made to the IPL procedure for activating IAM. The change will be to activate just the new production level of IAM. The old library, if it is still in the LINKLIST, can be removed. Then, after an IPL, the old IAM libraries can be deleted from the system.

In planning for the implementation, it is highly recommended that there be only one load library with any particular level of IAM. The primary reason for this recommendation is to avoid the problems encountered when there are multiple load libraries for the same version. The potential problems are differences in IAM Global Options settings, and maintenance concerns when applying zaps. If you do have to copy the IAM product from one library to another, be sure to delete the original library as soon as possible to prevent inadvertent use.

90.40 CONTINUED . . .

PREPARATION FOR PRODUCTION

The method of implementation for the first phase that will be least likely to cause problems is to perform an IPL to put the new version of IAM into production. Because many installations are not able to use this technique, there are alternative procedures provided below to implement the new version of IAM without an IPL. However, many of the tasks described below will have to be done even with the alternative procedures, so please be sure to review this methodology.

There are two steps to prepare for the implementation. First, is to revise the start up procedure(s) for IAM and second is to change the LINKLIST structure to include the new level of IAM. The changes to the startup procedure for IAM include adding a step to deactivate the old level of IAM after it is started, and if a TEST IAM is being used, change the JOBNAME value to JOBNAME=*, so that all jobs are eligible for the new version. Shown below is the revised multiple step PROC for starting Concurrent IAM, with a deactivation step inserted for the old version.

```
//IAMSTART  PROC  LIB='SYS1.IAM70.LOADLIB'
//* MEMBER(IAMSTART)
//******
//*    THIS PROCEDURE HAS BEEN CREATED TO START MULTIPLE
//*    VERSIONS OF THE IAM VIF.
//*    USER CHANGES:
//*      CHANGE 'SYS1.IAM70.LOADLIB' TO NAME OF OLD VERSION OF IAM
//*      CHANGE 'SYS1.IAM80.LOADLIB' TO NAME OF NEW VERSION OF IAM
//******
//******
//START70   EXEC  PGM=IAMSTART
//STEPLIB    DD   DISP=SHR,DSN=&LIB
//SYSLIB     DD   DISP=SHR,DSN=&LIB
//SYSUDUMP   DD   SYSOUT=*
//DEACT70   EXEC  PGM=IAMPARE            ← Deactivate old version
//STEPLIB    DD   DISP=SHR,DSN=&LIB
//SYSLIB     DD   DISP=SHR,DSN=&LIB
//START80   EXEC  PGM=IAMSTART
//STEPLIB    DD   DISP=SHR,DSN=SYS1.IAM80.LOADLIB
//SYSLIB     DD   DISP=SHR,DSN=SYS1.IAM80.LOADLIB
//STATS     EXEC  PGM=IAMSTATS
//STEPLIB    DD   DISP=SHR,DSN=SYS1.IAM80.LOADLIB
//SYSLIB     DD   DISP=SHR,DSN=SYS1.IAM80.LOADLIB
```

Figure 23: Example of PROC to Start Two IAM VSAM Interfaces and Deactivate One

The second step is to put the new IAM load library into the LINKLIST (in the appropriate LNKLSTxx or PROGxx member of SYS1.PARMLIB), and to optionally remove the old library from the LINKLIST. The alternative to changing SYS1.PARMLIB is to copy the new version of IAM into an existing LINKLIST library. That job should be run just prior to the IPL. If the plans are to copy the new IAM into the library where the old version resided, make sure that you have a copy of that version of IAM in another APF authorized library. If both the new and the old libraries are in LINKLIST, make sure that the new production version is in front of the old version.

After the IPL, the new version will be the production version. From the IAMSTATS, you will see that the old level of IAM is in storage, but INACTIVE. Should the need arise, the old version can be reactivated by running IAMSTART with that version's libraries as the STEPLIB and SYSLIB. Once the old version is reactivated, it can be accessed by JOBLIB or STEPLIB for selected jobs, or the library can be copied into LINKLIST in front of or over the new version, followed by an LLA refresh.

90.40 CONTINUED . . .

USING THE DYNAMIC LINKLIST

With OS/390 Version 1 Release 2 and z/OS, IBM has provided capabilities to dynamically change the system LINKLIST. This facility will make it possible to put the new version of IAM into production with minimal impact to the ongoing operation of the system. This procedure can also be used to put a new maintenance level of IAM into production, however you should run some tests prior to doing the cut over with running the new level of IAM with the old level of IAM.

1. Make sure that each version of IAM is in it's own unique library, and that each corresponding version of the IAM VSAM interface has been activated in the proper sequence with any other potentially interacting software products.

2. If the new IAM is a TEST IAM, set the JOBNAME=* so that it will process all jobs.

3. Issue the following Operator Commands:

   - **SETPROG LNKLST,DEFINE,NAME=NEWIAM,COPYFROM=CURRENT**
     This will build a new LINKLIST based on the currently active LINKLIST.

   - **SETPROG LNKLST,ADD,NAME=NEWIAM,DSNAME=newiam.loadlib,ATTOP**
     This command will place the new IAM library at the top of the LINKLIST concatenation, after SYS1.LINKLIB. Alternatively, instead of ATTOP, you can specify AFTER=dsname to place the new IAM library in a particular location. Be sure that if you do use AFTER= that the new IAM library will still be in front of the old IAM library.

   - **SETPROG LNKLST,DELETE,NAME=NEWIAM,DSNAME=oldiam.loadlib**
     This command is optional, it will remove the old IAM library from the new LINKLIST.

   - **SETPROG LNKLST,ACTIVATE,NAME=NEWIAM**
     This will activate the new LINKLIST for all new jobs and address spaces. Currently executing jobs will continue to use the old version of IAM.

4. Prior to performing an IPL, make sure to update the SYS1.PARMLIB with the LINKLIST changes, as well as changing the IAMSTART PROC if necessary.

When using the above procedure, do not deactivate the old version of IAM until you are certain that all jobs that were using that version of IAM have terminated. If you need to back out the new level of IAM, you can either activate the original LINKLIST, if you know the name, or build a new LINKLIST as the commands above do, deleting the new IAM library, and adding the old IAM library if you had deleted it. If you were running a TEST IAM where the prior IAM had the same internal level identifier, then deactivate the TEST IAM with the VIFSTOP JCL.

90.40 CONTINUED . . .

WITHOUT DYNAMIC LINKLIST The following procedure to put IAM into production without using the Dynamic Linklist facilities. Because this procedure does not utilize the dynamic LINKLIST, it will require a brief period of time where all activity against IAM files must be quiesced.

1. The first step is to make sure that there is an APF authorized backup library with the current production version of IAM. This must be separate from the target production LINKLIST library for the new version.
2. The second step is to make sure that both the current production version of IAM and the new version of IAM have been activated in the correct order regarding other software products in your installation. This is best accomplished by using one of the suggested multiple step IAM start procedures.
3. All activity on IAM files must be stopped. All open IAM files must be closed. When that point is reached, the following tasks are to be done:

   •Deactivate the production version of IAM using the VIFSTOP job.

   •Copy the new IAM library into the LINKLIST. The copy can be done into a library in front of the current production IAM library in LINKLIST, or into the current production library in LINKLIST. Make sure that the library that you copy the new IAM version into does not take any extents. If it does, a re-IPL will be necessary if running a compress does not get the dataset back down to it's original size.

   •Refresh the LINKLIST with the F LLA,REFRESH command.

   •If the new version is running as a TEST IAM, set JOBNAME=* by running the VIF START job with a PARM='TEST,JOBNAME=*'.
4. The new IAM is ready for use. IAM file activity can be resumed with the new production version of IAM.
5. When a back out procedure is no longer needed, revise the IAM start procedure for IPL to activate only the new production level of IAM. Then whenever the system is re-IPL'd, the new production version will be the only version of IAM installed on the system, and the backup library for the old version can be deleted.

If you need to backout the new IAM version, reactivate the old IAM, then copy the old IAM library back into the LINKLIST over the new version, and do an LLA Refresh. Once that is complete, you can deactivate the new IAM and resume processing of your jobs. Be sure to back out any changes you may have made to SYS1.PARMLIB for LINKLIST or the IAMSTART procedure.

FINAL STEPS The final phase of implementation of the new version or maintenance level of IAM can take place once you are satisfied with the performance and reliability of the product. To accomplish this task, change the start up of IAM during IPL to only start the new production version of IAM. If you are using the recommended multiple step PROC, then remove the start up and deactivation of the prior release. If the new IAM was being run as a TEST IAM, it can now be turned into a regular IAM by removing the parameter fields. (i.e. PARM='TEST,JOBNAME=*') The load library with the old IAM version can now be removed from the LINKLIST, and also from the APF list, presuming that the dataset was being used for IAM only. After the system is re-IPL'd, which can be done whenever it is convenient, then the new version of IAM is in full production. At this point, the old libraries can be deleted from your system.

THIS PAGE WAS INTENTIONALLY LEFT BLANK

91.01 GLOBAL OPTION CHANGE FACILITY - OVERVIEW AND JCL

The Global Option Change Facility gives the user a simple method of modifying installation options for executing the IAM system. The installation options include defaults for setting certain file attributes during DEFINE, enabling special features in IAM, and other processing options. This facility supplies the user with the ability determine what options they have changed with the AUDIT command, to PRINT present global option values, change values using the ZAP command, and RESET options to their original values as supplied on the installation tape. You should carefully review the options available for the IAM system.

BATCH EXECUTION To execute IAMZAPOP as a BATCH job use the following JCL:

```
//IAMZAPOP  EXEC  PGM=IAMZAPOP
//STEPLIB    DD   DISP=SHR,DSN=iam.loadlib
//SYSLIB     DD   DISP=SHR,DSN=iam.loadlib
//SYSPRINT   DD   SYSOUT=*
//SYSIN      DD   *

         put control cards here

/*
```

Figure 24: Example of JCL Skeleton to Execute IAMZAPOP

EXEC PGM=	Must specify the program name of the Global Option Change Facility - IAMZAPOP.
STEPLIB DD	Be sure that you run with the IAMZAPOP version that matches the Global Options Table that you are modifying by specifying a STEPLIB to the same library as the SYSLIB.
SYSLIB DD	Required DD which must specify the load module library in which the IAM Global Option Table resides.
SYSPRINT DD	Specifies the output message dataset. This is a required statement and usually is a SYSOUT dataset.
COPYTODD DD	Required DD only for the COPY operation, which specifies the load library containing the target option table for the COPY operation. The changed options in the SYSLIB option table will be copied to the target option table.
SYSIN DD	Required DD, which specifies the control statement dataset required for all functions. Usually an input stream or DD * dataset.

TSO EXECUTION The GLOBAL OPTION CHANGE FACILITY program (IAMZAPOP) can be executed under TSO. The program will prompt the user for the commands. 'END' will save the new options, if any, and terminate the program. The allocations required to execute IAMZAPOP in the TSO Foreground are as follows:

```
ALLOC   F(SYSLIB)     DA('iam.library')    SHR
ALLOC   F(SYSPRINT)   DA(*)
ALLOC   F(SYSIN)      DA(*)
IAMZAPOP
------- or if the IAM library is not in LINKLIST-------
CALL 'iam.library(IAMZAPOP)'
```

Figure 25: Invoking IAMZAPOP under TSO

91.02 GLOBAL OPTION CHANGE FACILITY - FORMAT AND OPTIONS

AVAILABLE FUNCTIONS The Global Option Change Facility has the following commands:

AUDIT Lists the Global Options that have been changed from their distributed defaults.

The AUDIT command format is: **AUDIT**

CANCEL Terminates IAMZAPOP without updating the Option Table.

The CANCEL command format is: **CANCEL**

COPY Copies the changed Global Options from the Option table in the SYSLIB dataset to the COPYTODD dataset. COPYTODD is a required DD statement for this operation.

The COPY command format is: **COPY**

END Terminates IAMZAPOP processing and rewrites the Option Table if any option was changed. This command is intended for TSO users.

The END command format is: **END**

HELP The HELP command will print or display a menu of the IAMZAPOP options and related documentation.

The HELP command format is: **HELP ALL**

PRINT If PRINT is specified, IAM will print or display the current values of the Global Options table.

The PRINT command format is: **PRINT**

RESET If RESET is specified, IAM will reset the Global Options to the original values on the installation tape.

The RESET command format is: **RESET**

ZAP Modify the Global Options. This command can be used to enable or disable specified processing options and set DEFINE defaults for the IAM system. The operands for this command are documented by function in the following sections.

The ZAP command format is: **ZAP operand=value, ...operand=value**

91.03 GLOBAL OPTION CHANGE FACILITY - ZAP COMMAND

This section discusses options that apply to most of the programs within the IAM system.

```
ZAP   [ BUFOPNO=nn]                        [,BUFSP=nnnnnnnn]
      [,CICSBUFSP=nnnnnnnn]                [,CORELIMIT=nnnnnn]
      [,CRBUFOPT= CYL | MCYL | MTRK | TRK]
      [,DATACOMPRESS=nnnnnn]               [,DATASPACE=nnnn]
      [,DESCRIPTCODE=nn]
      [,DISABLE=(BIM | ENHANCED | EURODATE | NOOWNID | NOREUSE |REORGWTO | VAM)]
      [,DSORG= DA| PS ]                    [,DYNCAT= YES | NO ]
      [,ENABLE=(BIM | ENHANCED | EURODATE | NOOWNID | NOREUSE | REORGWTO | VAM)]
      [,ESDSINTEGRATED=nn]
      [,INDEXSPACE= ALL | CICS | NO ]
      [,LIMITKEYS=nn ]                     [,LOADABWO= YES | NO ]
      [,MAXBUFNO=nn]                       [,MAXOVERFLOW=nnnnnn]
      [,MAXREGION=nnn]                     [,MAXSECONDARY=(x,y)]
      [,MINCOMPRESS=nn]                    [,MULTIVOLUME= PRIMARY | SECONDARY ]
      [,OCOREO%=nnn]                       [,OCOREX%=nnn]
      [,PE=nnnnn ]                         [,RECFM= F | V]
      [,RECTYPE=nnn]                       [,RELEASE= YES | NO ]
      [,RLS={SHARE1|SHARE2|SHARE3|SHARE4},{AND|OR},{TABLE}| NONE ]
      [,ROUTECODE=nn]                      [,SMF= YES | NO ]
      [,SORTCORE=nnnnnnn]                  [,SORTMSG=xx]
      [,SORTPFX=xxxx]                      [,STORCLASS=c......c]
      [,VAROVERFLOW=YES | NO ][,VSAMBLOCKF=n]
      [,VSAMWTO=YES | NO ]                 [,WORKDDNAME=xxxxxxxx]
      [,WORKPRIMARY=nnn]                   [,WORKSECONDARY=nnn]
      [,WORKUNIT=c......c]
```

91.03 CONTINUED . . .

OPERANDS Following are the operands for the ZAP command.

OPERAND	DESCRIPTION
BUFOPNO	Specifies a default minimum value for the initial number of buffers IAM will acquire when opening a file for processing. For batch jobs, IAM will acquire the larger of BUFOPNO, the number of blocks per track, or _ of the MAXBUFNO value. For CICS regions, IAM will acquire the larger of BUFOPNO or to the number of blocks contained on one track. The initial number of buffers IAM will acquire for a file can be overridden at execution time using the IAM Override Statement keyword MINBUFNO=. IAM's Real Time Tuning starts with this initial number of buffers and during file processing dynamically adjusts the number of buffers actually used for a file up or down based on demand. You may specify a value from 1 to 2048 buffers. The default is 4.
BUFSP	Specifies a default, in bytes, for the maximum amount of storage that IAM is to use for buffers when accessing a file in environments other then CICS. IAM divides this value by the file's block size to determine the number of buffers that will fit. For example: a file with 1/4 track blocking (13,682) on a 3390, IAM can fit 65 buffers in 875K of storage. IAM will use this value, or MAXBUFNO Global Option, which ever is higher, to set the maximum number of buffers for processing a dataset, unless overridden. BUFSP must be at least 65,536 bytes (64K). The default is 896,000 (875K), which will yield a value slightly higher than the number of buffers required to hold data from one cylinder.
CICSBUFSP	Specifies a default, in bytes, for the maximum amount of storage that IAM is to use for buffers when accessing a file under CICS. IAM divides this value by the file's block size to determine the number of buffers that will fit. For example: When a file with 1/4 track blocking (13,682) on a 3390, IAM can fit 19 buffers in 256K of storage. IAM will use this value, or MAXBUFNO Global Option, which ever is higher, to set the maximum number of buffers for processing a dataset, unless overridden. CICSBUFSP must be at least 65,536 bytes (64K). The default is 262,144 (256K).
CORELIMIT	Specifies the minimum prime index size required for IAM to consider using a compressed index structure. Any number from 0 to 999999, inclusive, may be specified. The default is 8000 bytes.
CRBUFOPT	Specifies the EXCP buffer option to be used during a file load process. The valid values are: **CYL** — Acquire enough buffers for one full cylinder. Each physical I/O (EXCP) is for one half of a cylinder. **MCYL** — Acquire enough buffers for two full cylinders. Each physical I/O (EXCP) is for a full cylinder. **TRK** — Acquire enough buffers for two tracks. Each physical I/O (EXCP) is for one track. **MTRK** — Acquire enough buffers for ten tracks. Each physical I/O (EXCP) is for five tracks. Default value is MCYL.

91.03 CONTINUED . . .

OPERAND	DESCRIPTION
DATACOMPRESS	Specifies the smallest size IAM file that will be considered for automatic data compression. This value is the minimum number of tracks that a DEFINE can specify as a file's primary allocation and still qualify for automatic data compression. Any number from 0 to 99999999, inclusive may be specified. The default is 75. IAM datasets that are defined as being 75 tracks or larger will default to being data compressed. **Innovation strongly recommends to keep Data Compression enabled.**
DATASPACE	Specifies the size, in megabytes, of the Data Space to be used for the temporary storage of the index to the IAM file that is being loaded. IAM will also use this value for the Index Space size, with a maximum Index space size of four times this value. Valid values are from 0 to 2048. A value of 0 results in the use of a dynamically allocated temporary dataset. Default is 256 megabytes.
DESCRIPTCODE	Specifies the descriptor code(s) to be used when issuing Write-To-Operator or Write-To-Operator-With-Reply messages. Any number from 1 to 16 inclusive may be specified. Multiple descriptor codes can be entered if specified as DESCRIPTCODE=(nn,...,nn). The default is 0 (X'0000').
DISABLE	Specifies the option(s) coded for this operand shall be deactivated. See the ENABLE operand for the options and their implications.
DSORG	Specifies the DSORG to be used when creating an IAM file. **DA** – Sets a DSORG of DA (direct access) **PS** – Sets a DSORG of PS (physical sequential) The default is PS.
DYNCAT	Specifies whether IAM should allow dynamic allocation to catalog the IAM file. This option is available to enhance IAM's support of BMC MAINVIEW (formerly POOLDASD). **YES** – Let dynamic allocation catalog file **NO** – IAM will catalog the file The default is NO.

91.03 CONTINUED . . .

OPERAND	DESCRIPTION
ENABLE	Specifies that the option(s) coded for this operand shall be activated. **BIM** – Enables IAM support for the BIM product. Default value is disabled. Requires a restart of IAM to take effect. **ENHANCED** – Specifies that IAM files will default to the Enhanced file format when they are defined. Default is that this option is enabled, which means that IAM files will default to the Enhanced format. **EURODATE** – Changes the format of date fields on IAMINFO and IAMPRINT reports to a European format of dd/mm/yyyy instead of mm/dd/yyyy. Default is disabled. **NOOWNID** – Causes IAM to save the actual value specified for OWNER on the define in the catalog entry for the OWNER field. If disabled, IAM stores it's own binary data in this field. The default is Enabled. **NOREUSE** – Causes IAM to honor the NOREUSE option if coded in the IDCAMS define statements. The default is Disabled, which sets all IAM datasets as reusable, except for IAM base clusters with an alternate index. **REORGWTO** – Causes IAM to display the IAMW22 messages when appropriate. To eliminate those messages, DISABLE this value. The default is Enabled. **VAM** – Enables IAM's CA-ALLOCATE (formerly SAMS / VAM) support. The default is Disabled.
<u>ESDSINT</u>EGRATED	Sets the INTEGRATED OVERFLOW percent for ESDS files. Reserves space in each block for record length changes. Recommended to be set to nonzero if you are using IAM ESDS files that are being updated. The default is 0.
<u>INDEXS</u>PACE=	Specifies the situations when IAM is to default to using a Data Space to hold the prime and overflow index for Enhanced format files. If IAM is using the Index Space, the original size used is the value set for the DATASPACE value in the Global Options Table, and a maximum size of four (4) times that value, up to 2048, will be set. Valid values are: **ALL** – All types of jobs that are accessing Enhanced Format files will use the Index Space. **CICS** – Only CICS regions will automatically use an Index Space. **NO** – Index Spaces will not be used automatically. Default is CICS.
<u>LIMI</u>TKEYS	Specifies the number of keys taken in a set when creating an IAM file with a compressed index. You may specify any number from 3 to 64, inclusive. The default is 32.

91.03 CONTINUED . . .

OPERAND	DESCRIPTION
LOADABWO	Specifies that IAMSTART (VIFSTART) will load the IAM interfaces for the DFP BWO and RLS Callable catalog information services. If you use a CICS VSAM recovery package that issues these calls, or are running with Transaction Server 1.2 or above, this option must be set to YES. Valid values are: **YES** – Enable the DFP Callable Services interfaces. **NO** – Do not enable the interfaces. The default is YES.
MAXBUFNO	Specifies the default maximum number of buffers IAM is permitted to acquire during file processing. IAM will use the higher of either this value, or the value for BUFSP. IAM's Real Time Tuning will dynamically adjust the number of buffers used for the file as demand warrants up to this maximum. You may specify a value from 1 to 2048 buffers. (Note that the maximum that will be used for Compatible Format files is 32.) The default is 5.
MAXOVERFLOW	For Compatible Format files, sets the maximum amount of overflow that will be allocated based on the CA% freespace value provided in the IDCAMS define. The default is 50,000.
MAXREGION	Specifies the default maximum value, in megabytes, that IAM will dynamically adjust the above the line Region value to. To disable the feature set value to 0. If your CICS regions are already at or above the 512 megabyte size, then increase this value. Default is 512.
MAXSECONDARY= (create,access)	Default multiplication factors for IAM Dynamic Secondary Space Adjustment feature. The first value is for file loads and the second value is for file updates. To disable this feature, set both values to 0. The Dynamic Secondary Space Adjustment feature is not used for IAM datasets that are DFSMS Extended Format. Default values are (10,5).
MINCOMPRESS	Specifies the minimum acceptable percentage of storage reduction achieved when creating an IAM file to determine if the file qualifies for a compressed index. May be any number from 8 to 40, inclusive. The default is 10.
MULTIVOLUME=	Specifies which space allocation value IAM will use when it appears that a dataset will take the next extent on the next volume for Enhanced format IAM files that are not DFSMS Extended Format. Valid values are: **PRIMARY** – Use the original primary allocation value when a volume switch is anticipated. **SECONDARY** – Use the original secondary allocation value when a volume switch is anticipated. Default is PRIMARY.

91.03 CONTINUED . . .

OPERAND	DESCRIPTION
OCOREO%	For Compatible format files only, specifies the amount of virtual storage for expansion of the Overflow index, as a percent of the total capacity of Independent Overflow, to be acquired when an IAM file is OPENed for update processing. 'nnn' may be any number from 1 to 100, inclusive. The default is 10.
OCOREX%	For Compatible format files only, specifies the amount of virtual storage for expansion of the Overflow index, as a percent of the total capacity of Independent Overflow, to be acquired when more memory is required. 'nnn' may be any number from 1 to 100, inclusive. The default is 10.
PE	For Compatible format files, specifies the number of blocks of Prime Extension area to be reserved when creating an IAM file. 'nnnnn' may be any number from 0 to 32767, inclusive. The default is 3.
RECFM	For Compatible format, non-data compressed files, specifies the internal record format IAM is to use for VSAM defines with equal average and maximum record lengths. **F** – Define the file as fixed. **V** – Define the file as variable. The default is F, however with the default of a file format of enhanced, the RECFM will always be V, unless the CREATE override of FIXED and COMPATIBLE is specified.
RECTYPE	Specifies the SMF 'user' record type to be written if SMF recording is requested for IAM files, nnn may be a number from 128 to 255, inclusive. There is no default value. This is a required field for SMF recording to be requested. Member IAMUSMF in the IAM Installation Control Library is a DSECT of the IAM SMF user record format. The suggested value is 201.
RELEASE	Specifies the default value for automatic release. **YES** – Unused disk space in an IAM file is to be released if Secondary allocation value is specified. **NO** – Unused disk space is not to be released. The default is YES.

91.03 CONTINUED . . .

OPERAND **DESCRIPTION**

RLS= Specifies the criteria IAM is to use to automatically determine which IAM files are eligible for IAM RLS processing. There is a combination of two forms of criteria available, which are by Share Options, and / or by dataset name. Valid values are:

AND – Indicates that to be automatically eligible for IAMRLS, a dataset must have the specified share options, must not be in the dataset name exclude list, and must be in the dataset name select list.

NONE – Indicates that no datasets are to be considered for automatic eligibility for IAMRLS. Set this option if you are not going to be activating the IAMRLS address space, or if you want to manually direct activity to IAMRLS through the IAM Overrides.

OR – Indicates that to be automatically eligible for IAMRLS, a dataset must have either the specified share options or be in the dataset name select list. If a dataset meets the share option eligibility, then neither the dataset name include or exclude lists will be examined. If a dataset does not meet the share option eligibility criteria, than it must not be in the exclude list, and must be in the include list. OR is the default if both TABLE and a SHAREx value are specified.

SHARE1 – Indicates that datasets with any cross-region share option value (1, 2, 3, or 4) will be eligible for IAMRLS processing.

SHARE2 – Indicates that datasets with cross-region share options of 2, 3, or 4 will be eligible for IAMRLS processing.

SHARE3 – Indicates that datasets with cross-region share options of 3 or 4 will be eligible for IAMRLS processing. This is the default option as shipped.

SHARE4 – Indicates that only datasets with a cross-region share option of 4 will be eligible for IAMRLS processing.

TABLE – Indicates that IAM is to search the dataset name include and exclude table to determine eligibility for IAMRLS processing, subject to other criteria. The default is that the dataset name tables will not take part in eligibility selection for IAMRLS.

For example, you could code: RLS=(SHARE3,AND,TABLE) which means files that are defined with share option 3 or 4 and are in the include dataset name table will be automatically eligible for IAM RLS processing.

Default values is SHARE3, files that are defined with share option 3 or 4 are automatically eligible for IAM RLS processing.

ROUTECODE Specifies the route code(s) to be used when issuing WTO of WTOR messages. Any number from 1 to 16, inclusive, may be specified. Multiple route codes can be entered as ROUTECODE=(nn,...,nn).

The default is 11 (X'0020').

SMF Specifies whether IAM is to write an IAM SMF user record when an IAM file is closed. The IAM SMF user record will contain all of the same information displayed in an IAM INFO Run Time Statistics Report. Member IAMUSMF in the IAM Installation Control Library is a DSECT of the IAM SMF user record's format.

YES – If a RECTYPE value is specified in IAM's GLOBAL OPTION TABLE an IAM SMF user record will be written whenever an IAM file is closed.

NO – IAM will not write IAM SMF user records.

The default is NO.

91.03 CONTINUED . . .

OPERAND	DESCRIPTION
SORTCORE	Specifies the amount of storage 'SORT' is to use for the IAM utility programs IAMRECVR and IAMSMFVS. You may specify any number from 10000 to 8000000, inclusive. The default is 100000.
SORTMSG	Specifies the message option to be used by the program 'SORT' if external sorting is required by IAMRECVR or IAMSMFVS. **AC** – all messages to the console **AP** – all messages to the printer (SYSOUT) **CC** – critical messages to the console **CP** – critical messages to the printer **NO** – no messages to be produced **PC** – critical messages to both console and printer The default is CC.
SORTPFX	Specifies the ddname prefix to be used by 'SORT' invoked by IAMRECVR and IAMSMFVS. If the string specified is less than 4 characters, a dollar sign ($) fill character will be used. The default is SORT.
STORCLASS	Specifies the SMS storage class to be used if SMS is active and a storage class was not specified on the define request, or the installation ACS routines did not assign an SMS storage class. If this option is blanks, no SMS storage group will be assigned. To reset this option to blanks, specify STORCLASS=' ' The default is blanks.
VAROVERFLOW	Specifies whether IAM is to use variable length overflow for IAM files when they are defined. Valid values are: **YES** – Set files as being eligible for variable length overflow when they are defined. **NO** – Do not set files as eligible for variable length overflow when they are defined. Default value is YES.
VSAMBLOCKF	Specifies the default blocking factor (number of blocks per track) to be used when an IDCAMS DEFINE does not specify CISIZE. Any value from 1 to 15 inclusive may be specified. The default value is 4 (four blocks per track).
VSAMWTO	For Compatible format files only, specifies the action to be taken under the VSAM interface, when an IAM file is not available. **YES** – An IAMW02 message is generated and the operator must reply Retry, Wait, or Cancel. **NO** – The Open is failed. The default is NO.
WORKDDNAME	Specifies the ddname of the work file used during an IAM load function. The work file will be used if IAM is not using a Data Space to hold the index. The default is IAMWKDD if it is in the JCL, otherwise the file will be dynamically allocated.

91.03 CONTINUED . . .

OPERAND	DESCRIPTION
WORKPRIMARY	Specifies the primary allocation in tracks of the work file dynamically allocated during an IAM load function. Any value from 1 to 65535, inclusive, may be specified. The default is 30 (tracks).
WORKSECONDARY	Specifies the secondary allocation in tracks for the work file dynamically allocated during an IAM load function. Any value from 1 to 65535, inclusive, may be specified. The default is 30 (tracks).
WORKUNIT	Specifies the unit name to be used when dynamically allocating the work file used during an IAM load function and during a DEFINE of an IAM dataset with the IAM ANYVOL support. The default is SYSDA.

91.04 IAMZAPOP JCL EXAMPLES

The following examples illustrate some of the ways of executing the GLOBAL OPTION CHANGE FACILITY.

EXAMPLE 1 The user wishes to display the present Global values.

```
//PRINT      EXEC  PGM=IAMZAPOP
//SYSPRINT    DD   SYSOUT=A
//SYSLIB      DD   DSN=iam.library,DISP=SHR
//SYSIN       DD   *
  PRINT
/*
```

Figure 26: Example of Printing out the IAM Global Options (EX9104A)

EXAMPLE 2 The user enables Automatic Data Compression for files that are 150 tracks or larger. The ZAP command will modify the IAM Option Table.

```
//ZAP        EXEC  PGM=IAMZAPOP
//SYSPRINT    DD   SYSOUT=A
//SYSLIB      DD   DSN=iam.library,DISP=SHR
//SYSIN       DD   *
  ZAP         DATACOMPRESS=150
/*
```

Figure 27: Example of Using IAMZAPOP to set Default for Data Compression (EX9104B)

EXAMPLE 3 Reset all of the Global Option values to their original values supplied on installation tape.

```
//RESET      EXEC  PGM=IAMZAPOP
//SYSPRINT    DD   SYSOUT=A
//SYSLIB      DD   DSN=iam.library,DISP=SHR
//SYSIN       DD   *
  RESET
/*
```

Figure 28: Example of Resetting IAM Global Options to Initial Values (EX9104C)

EXAMPLE 4 Enable the IAM SMF recording option so an SMF user record is written every time an IAM file is CLOSEd. IAM's SMF user record type is to be type 201.

```
//SETSMF     EXEC  PGM=IAMZAPOP
//SYSPRINT    DD   SYSOUT=A
//SYSLIB      DD   DSN=iam.library,DISP=SHR
//SYSIN       DD   *
  ZAP         RECTYPE=201,SMF=YES
/*
```

Figure 29: Example of Using IAMZAPOP to Enable IAM SMF Recording (EX9104D)

EXAMPLE 5 COPY all of the changed Global Option values to a new IAM load library.

```
//COPY       EXEC   PGM=IAMZAPOP
//SYSPRINT    DD    SYSOUT=A
//SYSLIB      DD    DSN=iam.library,DISP=SHR
//COPYTODD    DD    DSN=iam.new.library,DISP=SHR
//SYSIN       DD    *
  COPY
/*
```

Figure 30: Example of Using IAMZAPOP to Copy a Global Options Table (EX9104E)

91.04 CONTINUED . . .

EXAMPLE 6 Using the AUDIT command to list any Global Option values that have been changed from their distributed defaults.

```
//AUDIT      EXEC    PGM=IAMZAPOP
//SYSPRINT    DD     SYSOUT=A
//SYSLIB      DD     DSN=iam.library,DISP=SHR
//SYSIN       DD     *
  AUDIT
/*
```

Figure 31: Example of Using IAMZAPOP AUDIT to Determine Which Global Option Values Have Been Changed (EX9104F)

EXAMPLE 7 In this example, the RLS override is changed to automatically select those datasets defined with Share Option 3 or 4, or that are included in a dataset name table, that is provided to IAM RLS.

```
//SETRLS    EXEC   PGM=IAMZAPOP
//SYSPRINT   DD    SYSOUT=A
//SYSLIB     DD    DSN=iam.library,DISP=SHR
//SYSIN      DD    *
  ZAP RLS=(SHARE3,OR,TABLE)
/*
```

Figure 32: Example of Using IAMZAPOP to change the RLS automatic selection criteria (EX9104G)

THIS PAGE WAS INTENTIONALLY LEFT BLANK

READER'S COMMENT FORM
IAM USERS MANUAL V8.0
INNOVATION DATA PROCESSING

If you have comments on this manual, including:

- errors in the text or typographical errors
- clarity
- suggestions for improvement in the manual
- suggestions for improvement in the product
- any other comments

Please complete this form and fax it to Innovation at 973-890-7147 (in Europe you may fax it to your local Innovation office as shown on the front page of the manual).

You may also e-mail your comments to Innovation at support@fdrinnovation.com (be sure to identify the manual name in the message).

Your name: ____

Company name: ____

Mailing address: ____

____

E-mail address: ____

Comments:____

____

____

____

____

THIS PAGE WAS INTENTIONALLY LEFT BLANK